Mixed Autonomous Fleets in City Logistics

Von der Carl-Friedrich-Gauß-Fakultät

der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung des Grades eines

Doktors der Wirtschaftswissenschaften (Dr. rer. pol.)

genehmigte

Dissertation

von

Yannick Oskar Scherr

geboren am 10. Oktober 1992

in Pforzheim

Eingereicht am: 30. September 2020

Disputation am: 21. Dezember 2020

1. Referent: Prof. Dr. Dirk C. Mattfeld

2. Referent: Prof. Dr.-Ing. Bernhard Friedrich

3. Referent: Prof. Dr. Mike Hewitt

2021

Abstract

We consider a city logistics application in which a service provider seeks a repeatable plan to transport commodities from distribution centers to satellites. The service provider uses a mixed autonomous fleet that is composed of autonomous vehicles and manually operated vehicles. The autonomous vehicles are only able to travel independently on feasible streets of the heterogeneous infrastructure but elsewhere need to be pulled by manually operated vehicles in platoons. We introduce the service network design problem with mixed autonomous fleets to determine a tactical plan that minimizes the total costs over a medium-term time horizon. The tactical plan determines the size and mix of the fleet, schedules transportation services, and decides on the routing or outsourcing of commodities. We model this problem as an integer program on a time-expanded network and study the impact of different problem characteristics on the solutions. To precisely depict the synchronization requirements of the problem, the time-expanded networks need to consider narrow time intervals. Thus, we develop an exact solution approach based on the dynamic discretization discovery scheme that refines partially time-expanded networks containing only a fraction of the nodes and arcs of the fully time-expanded network. Further methodological contributions of this work include the introduction of valid inequalities, two enhancements that exploit linear relaxations, and a heuristic search space restriction. Computational experiments show that all evaluated variants of the solution approach outperform a commercial solver. For transferring a tactical plan to an operational solution that minimizes the transshipment effort on a given day, we present a post-processing technique that specifically assigns commodities to vehicles and vehicles to platoons. Finally, we solve a case study on a real-world based network resembling the city of Braunschweig, Germany. Analyzing the tactical and operational solutions, we assess the value of using a mixed autonomous fleet and derive practical implications.

Zusammenfassung

Wir betrachten eine Anwendung der urbanen Logistik, bei der ein Dienstleister einen wiederholbaren Plan für den Gütertransport von Distributionszentren zu Satelliten anstrebt. Dafür setzt der Dienstleister eine gemischt-autonome Flotte ein, die sich aus autonomen Fahrzeugen und manuell gesteuerten Fahrzeugen zusammensetzt. Die autonomen Fahrzeuge können nur auf bestimmten Straßen der heterogenen Infrastruktur selbstständig fahren, außerhalb dieser müssen sie von manuell gesteuerten Fahrzeugen mittels Platooning gezogen werden. Wir führen das "service network design problem with mixed autonomous fleets" ein, um einen taktischen Plan zu ermitteln, der die Gesamtkosten über einen mittelfristigen Zeithorizont minimiert. Der taktische Plan bestimmt die Größe und Zusammensetzung der Flotte, legt die Transportdienste fest und entscheidet über das Routing oder das Outsourcing von Gütern. Wir modellieren dieses Problem als ganzzahliges Programm auf einem zeiterweiterten Netzwerk und untersuchen die Auswirkungen verschiedener Problemeigenschaften auf die Lösungen. Um die Synchronisationsanforderungen des Problems präzise darzustellen, müssen die zeiterweiterten Netzwerke kleine Zeitintervalle berücksichtigen. Daher entwickeln wir einen exakten Lösungsansatz, der auf dem Schema des "dynamic discretization discovery" basiert und partiell zeiterweiterte Netzwerke entwickelt, die nur einen Teil der Knoten und Kanten des vollständig zeiterweiterten Netzwerks enthalten. Weitere methodische Beiträge dieser Dissertation umfassen die Einführung von Valid Inequalities, zweier Erweiterungen, die lineare Relaxationen verwenden, und einer heuristischen Suchraumbegrenzung. Experimente zeigen, dass alle evaluierten Varianten des Lösungsansatzes einen kommerziellen Solver übertreffen. Um einen taktischen Plan in eine operative Lösung zu überführen, die die Umladevorgänge an einem bestimmten Tag minimiert, stellen wir eine Post-Processing-Methode vor, mit der Güter zu Fahrzeugen und Fahrzeuge zu Platoons eindeutig zugeordnet werden. Schließlich lösen wir eine Fallstudie auf einem realitätsnahen Netzwerk, das der Stadt Braunschweig nachempfunden ist. Anhand der taktischen und operativen Lösungen bewerten wir den Nutzen einer gemischt-autonomen Flotte und leiten Implikationen für die Praxis ab.

Acknowledgements

I would like to thank everyone who contributed in some way to this dissertation and my journey towards finishing it. First of all, thank you to my committee members and to Markus Ludwig for chairing my defense. I would also like to acknowledge the support of the Deutsche Forschungsgemeinschaft for funding my research as part of the SocialCars research training group.

I am grateful for my supervisors and collaborators who have guided me and my research through good and bad times. Specifically, I would like to thank Dirk Mattfeld for his valuable advice, for always believing in me, and for providing me with freedom while working on this topic. I would like to thank Bernhard Friedrich for fruitful discussions and for providing different perspectives on my research. A special thank you goes to Mike Hewitt for his continuous support and advice. I particularly thank you for hosting me in Chicago twice which was not only helpful but also a great pleasure. In addition, I would like to thank Bruno Neumann Saavedra for his consistent support, patience, and friendship.

I want to thank all my colleagues and friends from both the Decision Support and the SocialCars group. Working with you has always been fun and inspiring. Finally, I would like to thank Ninja and my family for their unconditional love and support. I am deeply grateful to have you in my life.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

AV	Autonomous vehicle
CDC	City distribution center
CMND	Capacitated multicommodity network design
CTLPDP	Continuous-time load plan design problem
CTSNDP	Continuous-time service network design problem
DBCMND	Design-balanced capacitated multicommodity network design
DDD	Dynamic discretization discovery
IP	Integer program(ming)
LP	Linear program(ming)
LSP	Logistics service provider
MIP	Mixed-integer program(ming)
MV	Manually operated vehicle
NDD	Next-day delivery
SAE	Society of Automotive Engineers
SDD	Same-day delivery

SND Service network design

SNDAM Service network design with asset management

SNDMAF Service network design problem with mixed autonomous fleets

SNDRC Service network design with resource constraints

TSPTW Traveling salesman problem with time windows

Chapter 1

Introduction

With the rise of e-commerce, logistics service providers (LSPs) increasingly compete over shares in the parcel delivery market. Customers expect fast and reliable delivery which has prompted LSPs to offer next-day and same-day delivery options. With short lead times and small individual volumes of orders, however, LSPs struggle to consolidate orders and to operate efficiently. In densely populated urban environments, in which a large share of parcel delivery demand arises, congested roads and scarce parking opportunities provide additional burdens. One remedy is the consolidation of freight in city logistics concepts to reduce costs by maximizing the utilization of resources and, thus, minimizing the number of required resources. As LSPs also have a significant impact on traffic and the environment, municipal authorities promote city logistics concepts (Savelsbergh and Van Woensel, 2016).

A general framework to model the transportation of commodities in city logistics is provided by Crainic (2008). In the two-tier city logistics concept, satellites are introduced as intermediary transshipment locations. The first tier comprises consolidated transports from city distribution centers to satellites and the second tier comprises the delivery from satellites to customers. In general, tactical planning is used for determining a plan for a medium-term time horizon to efficiently utilize resources (Crainic and Laporte, 1997). In city logistics, tactical planning is primarily concerned with allocating resources, i.e., mainly vehicles, and scheduling repeatable transportation services in the first tier based on demand forecasts (Crainic et al., 2009). The optimization problem that aims to minimize the costs associated with these tactical decisions is denoted as the service network design (SND) problem (Crainic and Rousseau, 1986). SND problems are typically formulated on time-expanded networks to incorporate temporal information such as travel times between locations or time windows of commodities (Boland et al., 2019).

The profitability of an LSP is determined by the trade-off between guaranteeing a certain service quality and minimizing the operational costs for doing so. With a shortage of qualified drivers and steadily rising wages, labor costs constitute a large and growing share of the total costs. For this reason, the transportation industry views automated driving as an important upcoming technology with the potential to transform logistics and mobility services in the next years (Masoud and Jayakrishnan, 2017). Automated driving technology promises to control vehicles without a human driver in the vehicle which may significantly lower costs despite initial expenditure for hardware and software (Mosquet et al., 2015). As large fleet operators can achieve a high utilization of their vehicles while obtaining savings on labor costs, LSPs may be among the first to adopt and profit from automated driving (Mahmassani, 2016).

Whereas autonomous vehicles (AVs) have already been successfully deployed in different kinds of controlled environments such as warehouses, mining sites, or farmland (Heutger and Kückelhaus, 2014), public roads pose greater challenges in terms of safety and liability (Parker et al., 2017). Thus, fully autonomous vehicles, that can operate independently without a driver under any condition, are not expected to be on the market in the near future. The AVs we consider in this research are able to drive autonomously in specific driving modes which are determined by the surrounding infrastructure conditions. Dedicated lanes, easy-to-handle junctions, lower speed limits, or specific traffic rules may enable automated driving. Feasible roads are declared as AV zones by urban planning authorities, outlining a heterogeneous infrastructure in a city.

To travel on roads outside of AV zones, which are not feasible for automated driving, the AVs require human supervision. For this purpose, platooning can be used to link a group of vehicles consisting of one manually operated vehicle (MV) as a leader and one or more AVs as followers. Vehicle-to-vehicle communication ensures the propagation of control signals of the leading MV, which the following AVs replicate. Platooning has already been successfully deployed in field tests by several companies in the industry (Scania Group, 2018; Volkswagen AG, 2018; Volvo Trucks USA, 2018). LSPs in city logistics can particularly use platooning to transfer AVs between AV zones and to enhance the load capacity of a conventional MV. In the following, we denote a fleet composed of a mix of MVs and AVs as a mixed autonomous fleet.

The automobile manufacturer Renault (2018) envisions a similar approach in the futuristic EZ-PRO concept as depicted in Figure 1.1. Platooning of multiple vehicles near a distribution center is illustrated in Figure 1.1a. The handover process between an AV and a human, i.e., a delivery person or a customer, at a satellite is shown in Figure 1.1b.

(a) platooning (b) delivery

Figure 1.1: Renault EZ-PRO concept (Renault, 2018).

The overall objective of this thesis is to analyze the opportunities and limitations of using a mixed autonomous fleet in city logistics. For this reason, we consider an optimization problem of an LSP for the tactical planning of freight transportation in an urban environment and study efficient solution approaches for this problem. We particularly analyze the strategies the LSP may derive to utilize the two different vehicle types and to coordinate them in platoons.

In the following, we describe the structure of this thesis along with stating its contributions. Chapter 2 presents the background of the considered problem setting in literature and practical applications. In Chapter 3, we propose the service network design problem with mixed autonomous fleets (SNDMAF) for tactical planning and describe its features. With the SNDMAF, we contribute two innovative problem features to the service network design literature:

- the consideration of a mixed autonomous fleet of MVs and AVs that travel on a heterogeneous infrastructure,
- the utilization of platooning with MVs as leaders and AVs as followers.

In Chapter 4, we provide a mathematical model for this problem as well as a set of modifications to depict different operational policies of an LSP. We validate the functionality of the model in a computational study and assess potential cost savings of using AVs in Chapter 5. We particularly investigate the impact of the type of heterogeneous infrastructure, the demand pattern of commodities, and the fixed cost ratio between MVs and AVs on the tactical plan of the LSP. The problem, model, and parts of the computational study are also presented in Scherr et al. (2019). A similar problem setting along with preliminary results has been introduced prior in Scherr et al. (2018).

Chapter 6 is dedicated to the design and evaluation of efficient solution methods for the SNDMAF that are based on the dynamic discretization discovery scheme. The methodological contributions of this thesis are the following:

- adapting the dynamic discretization discovery scheme to the SNDMAF,
- introducing valid inequalities to strengthen the lower bounds,
- providing a new way to obtain feasible solutions in each iteration of the algorithm,
- enhancing the solution method with two exact variants that exploit relaxations,
- proposing a heuristic search space restriction to speed up the search for high-quality solutions.

The solution methods and experiments described in this chapter can in part also be found in Scherr et al. (2020).

In Chapter 7, we provide a post-processing technique to operationalize a tactical plan. Based on a flow-based SNDMAF solution, the post-processing technique generates feasible routes for each MV and AV including the assignment of commodities to individual vehicles while minimizing the transshipment effort. In Chapter 8, we analyze SNDMAF solutions together with respective operational solutions in a case study based on a real-world network for the city of Braunschweig. Since the heterogeneous infrastructure in this network is based on a pilot study for automated driving, the case study enables to assess the deployment of a mixed autonomous fleet in an existing infrastructure to derive practical implications. In Chapter 9, we conclude and provide perspectives for future research.

Chapter 2

Background

In this chapter, we provide a review on literature and current practice related to the problem setting considered in this dissertation. The first four sections of this chapter cover different planning-related and technological fields. In Section 2.1, we distinguish between the typical planning levels in transportation and particularly focus on service network design problems for tactical planning. In Section 2.2, we describe the concept of city logistics and provide real-world applications. Sections 2.3 and 2.4 are dedicated to autonomous vehicles and platooning, for both of which we report technological advancements and operations research literature. Finally, we identify and describe the research gap that the work presented in this thesis targets in Section 2.5.

2.1 Transportation planning

We first define the three widely used planning levels of operations research and particularly apply their description to the area of freight transportation in Section 2.1.1. Then, we focus on tactical planning and give an introduction to SND problems that are commonly used for transportation applications in Section 2.1.2. In Section 2.1.3, we describe the concept of time-expanded networks in more detail. We describe resource management considerations in Section 2.1.4 before we state relevant formulations and solution approaches for SND problems in Section 2.1.5.

2.1.1 Planning levels

Logistics planning is typically divided into three levels that are hierarchically linked to each other (Crainic and Laporte, 1997). In the following, we define the planning levels in general and present examples for each of the levels that specifically concern the planning of freight transportation operations of an LSP.

In Table 2.1, we characterize the decisions in each planning level according to three distinct characteristics. The *horizon* describes the duration that is considered for the decision and indicates the longevity of its effects on the future. The *resolution* describes the precision of individual decisions and their impact. The *scope* describes how comprehensive and far-reaching a decision is on the overall activities. Based on these characteristics, we distinguish between the strategical, tactical, and operational level of planning.

Table 2.1: Decisions and their characteristics in different planning levels.

Planning level	Horizon	Resolution	Scope	Example
strategical	long-term	coarse	large	facility location planning
tactical	mid-term	medium	medium	service network design
operational	short-term	fine	small	vehicle routing

Strategical level Strategical decisions are long-term decisions made with a large and far-reaching impact on future activities. Due to the large horizon of one or multiple years considered, the resolution is typically coarse such that the impact of one individual decision is hard to determine precisely. The acquisition of resources falls into this category such as building facilities, buying vehicles, or hiring staff. An example in freight transportation would be the facility location planning of distribution centers and transshipment terminals.

Tactical level Tactical decisions are located between strategical and operational decisions in their horizon, resolution, and scope. The considered time horizon is generally shorter than a year, e.g., a season, month, or week. The complete planning horizon is typically discretized into time periods to yield a medium resolution. The scope of decisions in the tactical level is smaller than in the strategical level as some conditions are already fixed by strategical decisions. Tactical decisions concern the allocation of resources as well as the planning of repeatable tasks that rely on regularity. As an example for tactical planning, an LSP may assign vehicles and staff to distribution centers and schedule master tours that may be repeated over a given planning horizon.

Operational level Operational decisions are subordinate to the guidelines provided by preceding strategical and tactical decisions which gives them a narrower decision space. A typical time horizon contains one day or even a work shift which is mostly depicted in continuous time, allowing for fine resolution and decisions that impact activities precisely. Operational decisions are related to activities that are directly executed by resources or workers in the short term. Examples in a freight transportation context are the daily routing of vehicles from customer to customer as well as the packing of commodities in vehicles.

2.1.2 Service network design problems for tactical planning

Service network design (SND) problems originate from network design problems that are widely studied in a variety of applications. Magnanti and Wong (1984) provide a general model for network design in transportation. In the capacitated multicommodity network design (CMND) problem, commodities must be routed between different facilities over a network with limited capacity. Typically, a fixed cost needs to be paid to open an arc in the network. The selection of opened arcs determines a design that can be used for routing the commodities considering variable costs. Network design problems generally address strategical planning issues with a long-term planning scope.

SND problems are first introduced by Crainic and Rousseau (1986) for freight transportation planning. They address the tactical planning level that regards a mid-term planning scope and focuses on the efficient use of assets on a regular basis. To this end, the problem can be divided into two components. The first component is the design of so-called services that need to be selected in a manner similar to arc selection in network design problems. These services typically represent transportation legs or jobs that are operated by a service provider using resources such as vehicles. The costs that are caused by the operation of those services are typically considered as fixed costs in the SND problem. The entirety of the installed services constitutes the so-called service network which provides transportation capacities on a regular basis.

The second component of SND problems is the routing of commodities from their origin to their destination through the service network. To this end, the commodities, which typically feature a specific volume or weight, need to be coupled with the services while respecting the transportation capacities. The variable costs that are caused by the transportation of commodities may be based on the additional weight of the load or the operational efforts for loading the commodities. This problem component of assigning commodities to services with limited capacity shows similarities to knapsack problems.

SND problems have been applied in multiple and diverse fields in the area of transportation. The studied applications include less-than-truckload (Crainic, 2000), express shipment (Barnhart et al., 2002), and ferry (Lo et al., 2013) networks as well as shared mobility systems (Neumann-Saavedra et al., 2020), among others. We refer to Crainic (2000) and Wieberneit (2008) for literature reviews on SND problems and their different applications in the area of transportation.

2.1.3 Time-expanded networks

The first SND problems, similar to most network design problems, were modeled on static networks that consider the spatial location of nodes or some kind of logical structure. However, with more and more temporal attributes coming into play, such as departure and arrival times for commodities, travel times for vehicles, and shift lengths for staff, researchers tended to specifically model time as an attribute in the underlying network of the problem. The term *scheduled* service network design is also used for these kinds of problems to emphasize that time plays a dominant role. In time-expanded networks, also called time-space networks, time is explicitly modeled by replicating locations in discrete time periods.

Researchers have been using time-expanded networks to model flow problems in a more efficient manner for decades. Ford and Fulkerson (1958) convert a maximum flow problem over time on a physical network, that considers travel times on the arcs, to yield a static flow problem on a time-expanded network. Solutions to the problem are depicted using so-called dynamic flows from a source to a sink. Skutella (2009) provides a more recent introduction to different network flow problems over time, points to relevant literature, and defines the properties of time-expanded networks.

Time-expanded networks are generated on the basis of a physical network and a time horizon. In Figure 2.1, we illustrate how the locations of a physical network are expanded in time to yield a time-expanded network for a time horizon of length t_{max}. To this end, the location is depicted on the y-axis and the time is depicted on the x-axis of the time-expanded network. The arcs of the time-expanded network connect the nodes in different time periods to depict traveling between locations. The length of the arcs, that we measure based on the x-axis, equals the weight of the arcs of the physical network, that represents the travel time between locations. Thus, the arcs in a time-expanded network can be interpreted in the following way. The origin of an arc depicts the departure time at an origin location and the destination of an arc depicts the arrival time at a destination location.

By considering time explicitly in the network, we obtain a static flow problem which is generally more tractable to solve (Skutella, 2009). This is due to the fact that it does not require big-M constraints that are typically used to model the temporal sequence of node visits as in, e.g., typical vehicle routing formulations (Vigo and Toth, 2014). However, the discretization of a time-expanded network, i.e., the length of its time intervals, is a crucial parameter. Boland et al. (2019) study the impact of discretization on the obtained solution

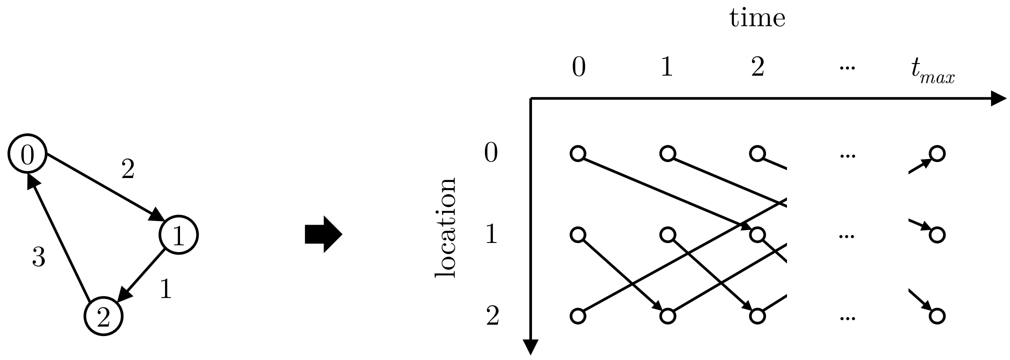


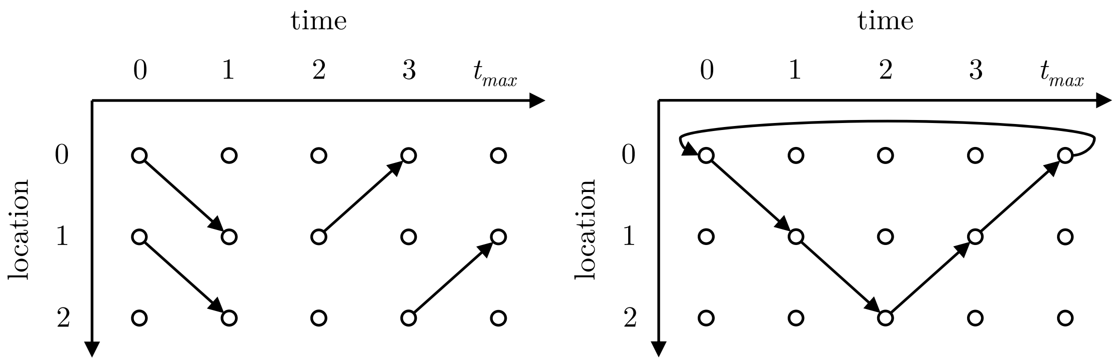
Figure 2.1: Building a time-expanded network from a physical network.

quality and the computational tractability of an SND problem. Fine discretization, i.e., using small and many time intervals, leads to large time-expanded networks that can depict time-dependent decisions accurately in the solution. Coarse discretization, i.e., using large and few time intervals, leads to small time-expanded networks with a larger tolerance in terms of the solution accuracy. In general, the discretization is determined based on problem characteristics and an acceptable trade-off between accuracy and computational effort. Often, however, practical considerations can impact the choice of discretization, such as company policies or the accuracy of the data based on which the models are built.

2.1.4 Resource management considerations

Over the recent years, the general SND problem was enriched to integrate additional resource management considerations that occur in practice. By depicting the operational effects more precisely in tactical planning, the tactical plan can be implemented with minor modifications in the operational level. One such operational consideration is the allocation and repositioning of resources, which may comprise vehicles, power units, or staff, among others. For this reason, Pedersen et al. (2009) formulate the design-balanced capacitated multicommodity network design (DBCMND) problem. This formulation includes so-called design-balance constraints that ensure the availability of a resource at a node by balancing the incoming and outgoing services, thus resembling flow conservation constraints. Andersen et al. (2009) expand this idea with the introduction of service network design with asset management (SNDAM). Here, the assets or resources that are available need to follow cyclic routes. In this way, the services are connected such that they can be executed repeatedly while considering the transition between two consecutive time horizons. The selection of an asset is associated with fixed costs that typically predominate the service costs. The authors also introduce constraints to limit route lengths.

In Figure 2.2, we illustrate a simple example to show the impact of resource management considerations on a tactical plan for an SND problem. For the sake of simplicity, we assume that the underlying physical network for this example is a complete graph with arc weights of one, which is why each arc in the time-expanded network has a length of one time period.



(a) without resource management (b) with resource management

Figure 2.2: Impact of resource management considerations on a tactical plan.

In Figure 2.2a, the services of an SND solution *without* resource management considerations are depicted. The arcs in the time-expanded network represent the installed services. We observe that the planned services could not be performed by one vehicle as it cannot be at different locations at the same time. Thus, the tactical plan can only be implemented with an additional vehicle to yield an operational solution. Figure 2.2b shows a solution *with* resource management considerations. The spatial origins and destinations of the services are the same as in the example without resource management, but the dispatch times are adapted. Design balance is ensured at each node of the time-expanded network and the services are connected to form a cycle. With this solution, one vehicle is able to perform all services while starting and ending at the same location.

Further enhancements to the general SND problem have been developed that integrate strategical decisions concerned with resources. Crainic et al. (2016) expand the formulation and introduce service network design with resource constraints (SNDRC). The resource bound constraints explicitly limit the number of resources available at a facility. Crainic et al. (2017) build upon these extensions but handle multiple types of resources with different features in their scheduled service network design problem with resource acquisition and management. This formulation also includes a strategic layer that allows for the acquisition of resources as well as their assignment and potential reassignment to a facility. Hewitt et al. (2019) recently extend the model to account for demand uncertainty. Wang and Qi (2019) consider an SND problem for an integrated transportation system with multiple types of services.

2.1.5 Formulations and solution approaches

To model SND problems on time-expanded networks, there exist different kinds of formulations based on mixed-integer programming (MIP). Originally, arc-based formulations are used similar to the general CMND formulation. With the introduction of resource management considerations to SND, alternative formulations have been proposed. For this reason, two groups of decision variables, design variables and commodity flow variables, can be distinguished in those models. As commodities typically have an origin and a destination, each feasible path they can follow may be represented by a path variable. Resources, e.g., vehicles, may also follow paths if their purpose is to carry commodities. If there are stronger requirements in terms of managing resources, e.g., vehicles have to start and end from a specific facility, it may be more reasonable to directly consider cycles as design variables.

Andersen et al. (2009), for example, present four different representations of their SNDAM model in which design variables are defined for arcs or cycles and commodity flow variables are defined for arcs or paths. However, these paths and cycles need to be enumerated a priori and each of them is represented by an integer variable. Thus, the large number of variables leads to models that are not tractable for off-the-shelf solvers without adapting techniques that consider only a subset of variables, such as column generation (Crainic et al., 2016).

Solution approaches for SND problems comprise different exact and heuristic methods. State-of-the-art MIP solvers are generally able to solve smaller instances to optimality or within a small optimality gap. However, their performance is not only impacted by the size of the physical network or the number of commodities but largely by the length and discretization of the planning horizon (Boland et al., 2019). Another exact approach is the branch-and-price algorithm that Andersen et al. (2011) propose for the SNDAM with cycle design and path flow variables. Recently, the dynamic discretization discovery (DDD) algorithm was proposed by Boland et al. (2017) as an exact method to solve the so-called continuous-time SND problem, which does not require to determine the time discretization a priori. We review the DDD and related algorithms in Section 6.1 in more detail before we adapt the DDD scheme to solve the SNDMAF.

A number of heuristic approaches have also been proposed to solve SND problems in different variants, among them several metaheuristics. Pedersen et al. (2009), for example, propose a tabu search approach to solve the DBCMND. Matheuristics constitute another category of solution approaches that is increasing in popularity. Those enhance the usage of a MIP solver with heuristic elements such as neighborhood search and learning mechanisms. Chouman and Crainic (2015) present a cutting-plane matheuristic for the DBCMND, while Crainic et al. (2016) use slope scaling and column generation techniques to solve the cycle-based formulation of the SNDRC problem. In general, the type of formulation of the model can narrow down the choice of a promising solution method. Since there exist multiple variants of SND problems that each may be modeled in different ways, the large number of resulting problem-model-method combinations still leaves room for further research.

2.2 City logistics

In this section, we motivate the challenges of city logistics in Section 2.2.1 and particularly describe the two-tier city logistics concept in Section 2.2.2. Finally, we review planning problems that have been studied for city logistics applications in more detail in Section 2.2.3.

2.2.1 Motivation

City logistics pursues effective and efficient transportation of goods in urban areas (Savelsbergh and Van Woensel, 2016). While service providers in transportation typically aim to minimize individual costs, city logistics further recognizes the impact of their operations on traffic and the environment (Taniguchi and Thompson, 2001). Since urban population continues to grow and cities become more crowded, the challenge to consider those negative effects and still provide a profitable service becomes even more difficult (Taniguchi and Thompson, 2014). Furthermore, e-commerce growth and the trend towards offering same-day delivery continuously lead to an amplification of the challenges. These issues keep researchers busy to this day, having led to a variety of different frameworks to optimize city logistics services. A common objective in most of these model applications is the reduction of freight vehicles in the city and their distance traveled.

There have been numerous city logistics projects with multiple stakeholders undertaken in practice, some of them more successful than others. While some of these projects are mainly initiated by collaborating private companies, public authorities support others through subsidies or policies that favor participating companies. Benjelloun et al. (2010) investigate city logistics projects and classify their main operating principles through five aspects:

- *Consolidation*: City logistics systems extensively benefit from consolidating loads into the same vehicles. Multiple service providers may also consolidate their loads at urban consolidation centers.

- *Modal shift*: The negative impacts of road vehicles can be resolved by transferring freight to other modes, such as rail or water.
- *Regulation*: City authorities can influence the organization and management of freight transportation using incentives or restrictions. Regulatory measures may concern temporal or spatial restrictions, environmental standards, or access charges.
- *Intelligent transportation systems*: City logistics systems can be improved by obtaining, processing, and distributing information related to traffic via intelligent transportation systems.
- *Cooperation*: Different city logistics service providers may cooperatively manage demand, fleets, or other services beyond transportation.

Several stakeholders such as the city, government, shippers, carriers, citizens, or universities may be involved in such projects. Sophisticated planning and evaluation models need to be applied to reconcile the mentioned diverse aspects and to incorporate the interests of each stakeholder.

2.2.2 Two-tier city logistics

According to Crainic (2008), consolidation and coordination are the fundamentals for successful city logistics planning. Consolidation can be achieved if various shipments are pooled and assigned to a single vehicle. Coordination is encouraged by considering the fleets of multiple service providers. One commonly used approach to incorporate both of these aspects introduces city distribution centers (CDCs) on the city's periphery. In single-tier city logistics, incoming loads from long-haul transportation are unloaded, sorted, and consolidated into smaller trucks that deliver them to urban destinations. Inbound flow of goods represents the largest share of load volume in most cities, therefore the majority of city logistics literature focuses on inbound flow only. Nevertheless, outbound flow and shipments within the city can be considered to further improve consolidation in cities (Crainic et al., 2012).

Two-tier city logistics is introduced by Crainic (2008) and describes an advancement of the single-tier case. We depict a possible setting in Figure 2.3. Satellites are introduced as transshipment terminals, dividing freight distribution into a first and a second tier. These satellites are typically conceptualized as spaces that allow for vehicles of the first tier and the second tier to meet and transship goods. Crainic et al. (2009) further note that "no permanent facility thus needs to be constructed for satellites, existing urban spaces, e.g., city squares and parking lots, being used instead". Since no warehousing facility is available, storage of goods is not provided.

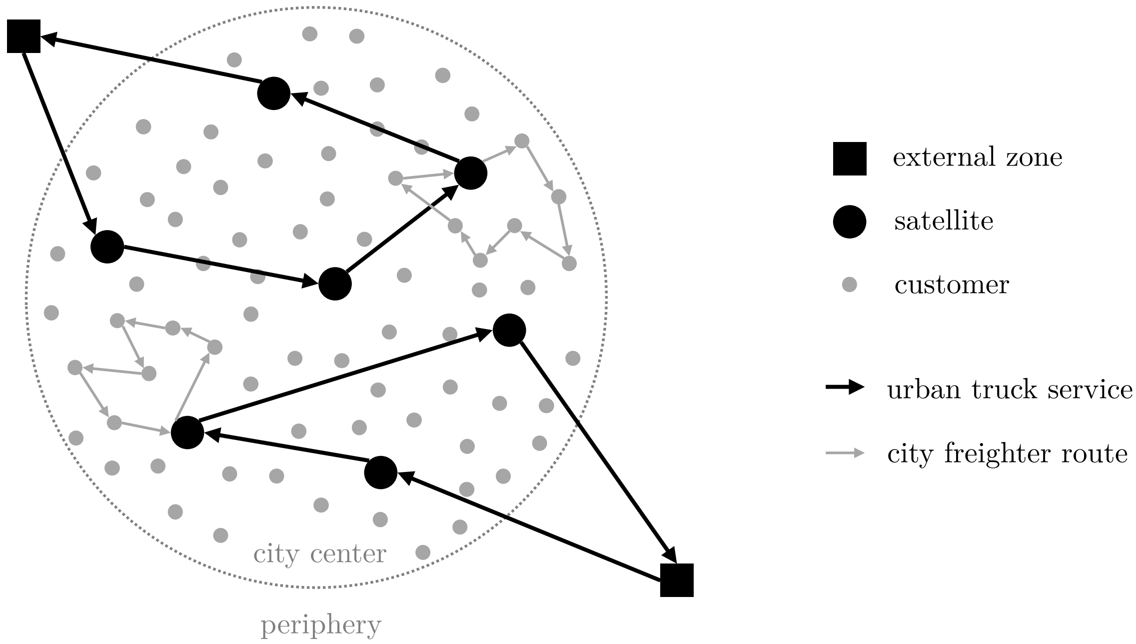


Figure 2.3: Two-tier city logistics setting.

In a conventional inbound system, goods appear in external zones outside the city center, where nearby highways, ports, or airports enable the arrival of high-volume shipments. CDCs as part of external zones act as consolidation hubs for multiple service providers and pool incoming goods. Urban trucks are used for the transport to inner-city satellites in the first tier. Their routes preferably take place in dedicated corridors, e.g., "ring highways", taking traffic and emissions into consideration. In the second tier, city freighters usually cover the last mile delivery between satellites and customers. These vehicles with smaller load capacities are able to travel on any given street in the city while emitting less noise and exhaust gas than urban trucks. In the example given in Figure 2.3, we also depict the services of two urban trucks and the routes of two city freighters.

2.2.3 Planning problems

Researchers have investigated several aspects of city logistics that can be grouped based on the hierarchical planning levels described in Section 2.1.1. Strategical planning problems mostly concern the location of CDCs and satellites that are typically determined by solving location-routing problems. Gianessi et al. (2016), for example, study the multi-commodity ring location routing problem that seeks to locate CDCs and connects them via a ring on which goods are transported. Crainic et al. (2010) analyze the location of satellites with a two-echelon vehicle routing problem. Franceschetti et al. (2017) study a strategic fleet planning problem for determining the fleet mix by solving an area partitioning problem in which the route lengths are calculated using continuous approximation.

Tactical planning focuses on an efficient allocation of resources and aims to enhance consolidation of transports. Crainic et al. (2009) develop a two-tier city logistics framework that consists of an SND problem in the first tier and a vehicle routing problem in the second tier. Considering both tiers in an integrated optimization problem yields a complex model that is difficult to apply in practice, which is why the tiers are usually decomposed. The first-tier problem is further described by Crainic and Sgalambro (2014) as the urban-vehicle service network design problem. While the model ensures design-balance of vehicles in the so-called uncontrolled variant of the model, the controlled variant additionally includes resource management considerations.

For operational planning in city logistics, vehicle routing problems are typically considered to determine the sequence of customer visits (Cattaruzza et al., 2017). Taniguchi and Thompson (2002) propose a model for stochastic vehicle routing and scheduling. Routing problems can be further enriched to consider practical considerations such as for example time-dependent travel times (Ehmke et al., 2012). Hemmelmayr et al. (2012) among others consider a two-echelon vehicle routing problem to cover the two tiers of city logistics in one integrated model. However, fleet management decisions concerned with acquiring or assigning vehicles are generally not considered in the operational planning level.

Recent innovative contributions to the research area include multi-modal transportation via line-based services, such as trams with cargo compartments, in addition to road-based services (Fontaine et al., 2017). The authors also integrate inbound demand from external zones to satellites and outbound demand from satellites to external zones within one model to improve load utilization of vehicles. Masson et al. (2017) propose a two-tier city logistics system with mixed transportation of passengers and goods.

Further, the vision to progress towards hyperconnected city logistics was presented recently, which combines ideas of city logistics and the physical internet (Crainic and Montreuil, 2016). Here, standardized containers that store commodities, so-called π-containers, enable a more seamless movement of commodities through a network of openly available hubs. Crainic et al. (2020) particularly investigate coalitions of LSPs that share their resources and information flows in such a system.

2.3 Autonomous vehicles

In the following, we provide relevant background on automated driving technologies and AVs. In Section 2.3.1, we motivate the development and use of AVs. Then, we describe technological advancements and a classification for automated driving in Section 2.3.2. Finally, different logistics applications that consider AVs and the respective planning problems are presented in Section 2.3.3.

2.3.1 Motivation

Driving automation is a technology to which a lot of research has been dedicated in recent years. The overall aim of the research in this area is to develop fully-automated or autonomous vehicles (AVs) that do not require any input from a human driver at all. Therefore, existing driver assistance features such as adaptive cruise control have been enhanced to take over more and more tasks from the driver. To progress further, additional hardware and software such as more advanced sensors and image processing techniques still require further research. Although AVs are not yet on the market, several companies and research institutes are actively testing AV prototypes at advanced stages of development.

The legal foundation and the operational implications of driverless vehicles are still vague. Although automated driving technologies have already been successfully used in controlled industrial environments (Heutger and Kückelhaus, 2014), their use on public roads still causes discussion of safety and liability issues (Parker et al., 2017). Equipping a vehicle with the necessary hardware and software for automated driving, particularly sensor technology, comes at a one-time cost. At launch, market prices for purchasing an AV are expected to be around 10,000 $ higher than a comparable vehicle without such technology (Mosquet et al., 2015). Due to the high utilization of their fleet, commercial fleet operators like LSPs may become early adopters in terms of AV deployment and profit from the technology first (Mahmassani, 2016).

A large area of research is dedicated to study the impact of AVs on traffic flow stability and throughput. Talebpour and Mahmassani (2016) show in a simulation study that higher market penetration rates, i.e., a larger share of AVs among all road vehicles, can increase overall throughput. Other studies investigate possibilities to increase this penetration rate by making utilization of AVs more economical. Masoud and Jayakrishnan (2017) propose a shared ownership program among clusters of households and study the expected benefits.

2.3.2 Technological advancements

Different levels of driving automation can be distinguished that are in part already available and in part projected for the near future. The US-based Society of Automotive Engineers (SAE) provides a guideline with standard J3016, in which six SAE levels of driving automation are defined (SAE International, 2018). We depict these levels and their description according to the SAE standard in Figure 2.4.

SAE level	Name	Narrative Definition	Execution of Steering and Acceleration/ Deceleration	*Monitoring* of Driving Environment	Fallback Performance of *Dynamic Driving Task*	System Capability *(Driving Modes)*
***Human driver* monitors the driving environment**						
0	No Automation	the full-time performance by the *human driver* of all aspects of the *dynamic driving task*, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a
1	Driver Assistance	the *driving mode*-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	Human driver and system	Human driver	Human driver	Some driving modes
2	Partial Automation	the *driving mode*-specific execution by one or more driver assistance systems of both steering and acceleration/ deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	**System**	Human driver	Human driver	Some driving modes
***Automated driving system* ("system") monitors the driving environment**						
3	Conditional Automation	the *driving mode*-specific performance by an *automated driving system* of all aspects of the dynamic driving task with the expectation that the *human driver* will respond appropriately to a *request to intervene*	System	**System**	Human driver	Some driving modes
4	High Automation	the *driving mode*-specific performance by an automated driving system of all aspects of the *dynamic driving task*, even if a *human driver* does not respond appropriately to a *request to intervene*	System	System	**System**	Some driving modes
5	Full Automation	the full-time performance by an *automated driving system* of all aspects of the *dynamic driving task* under all roadway and environmental conditions that can be managed by a *human driver*	System	System	System	**All driving modes**

Figure 2.4: Levels of driving automation following standard J3016 (SAE International, 2018).

Level 0 provides no automation, whereas Levels 1 and 2 assist the driver partially. While such systems greatly contribute to traffic safety and convenience for the driver, their economic impact on industrial applications is negligible as long as driving staff is still needed. From Level 3 upwards, an automated driving system monitors the driving environment, thus performing the entire dynamic driving task. In conditional automation (Level 3), the human driver still serves as a fallback to take back control from the system. In addition, the automated driving system is only capable of some driving modes like highway cruising, low speed traffic jams, or operations in closed areas. In Level 4 (high automation), the fallback is provided by the system. Level 5 ensures full automation in all driving modes.

Since fully autonomous driving, especially in cities, is not expected in the next few years, a more realistic scenario is that vehicles can drive autonomously under specific and carefully evaluated driving modes (CARMERA, 2018). This scenario would fall into Level 4 in which an autonomous driving system can react to fallbacks without human intervention. Here, the automated system's ability to control a vehicle mainly depends on the infrastructure conditions surrounding it. Integrating AVs into a city environment may require costly infrastructure preconditions both to ensure traffic safety and to advise traffic flows in real time. Also, 3D mapping of complex street networks may face difficulties in specific geographic areas, establishing so-called "geofences" around them. Thus, the deployment of AVs is delayed within these parts of the network (Coker, 2018; Hook, 2018). The design of dedicated AV zones, where AVs are especially desired by traffic authorities, is evaluated by Chen et al. (2017). Li et al. (2020) study the deployment of road-side units to provide AVs with beyond-line-of-sight information on their environment.

2.3.3 Logistics applications

At this point, the effects of automated driving in logistics are hard to predict. Therefore, industry and academia consider several scenarios and use cases. Chottani et al. (2018) predict that autonomous trucks for long-haul transportation on highways will likely roll out in waves. Applications in the near future concern platooning as a way to automate the following trucks on interstate highways. In a short transition period, trucks may operate driverless on highways but drivers may be required to drop off trucks at dedicated truck stops. From 2027 on, the study expects trucks to drive fully autonomously everywhere.

Also, AVs are being tested in several transportation applications outside of highways. The startup Gatik deploys autonomous vans with human safety operators as backup on Walmart's "middle mile", i.e., the delivery from warehouses to stores (Davies, 2019). The vans operate on a set of known routes that are feasible for autonomous driving. In last mile delivery, DHL imagines self-driving parcel stations to play an important role (Heutger and Kückelhaus, 2014). The delivery robot company Nuro recently received federal approval in the United States to deploy its R2 model on public streets at a maximum speed of 40 km/h (Duncan, 2020). Startups like Starship Technologies (2020) deploy smaller robots, that typically have a load capacity of a single parcel, on sidewalks. However, these robots can face problems when encountering obstacles or interacting with humans (Bukszpan, 2019).

In the operations research literature concerned with automated delivery applications, there is a strong focus on unmanned aerial vehicles or drones. Several of the studies in this field consider truck-and-drone routing problems in which a truck carries a number of drones that are able to take off from the truck's roof, perform a delivery, and meet with the truck again. Murray and Chu (2015), Poikonen et al. (2017), Agatz et al. (2018), and Wang and Sheu (2019) state this problem in similar ways that differ in the number of trucks and drones considered. Poikonen and Golden (2019) expand the formulation to consider multiple visits of a drone between take-off and landing. Boysen et al. (2018b) adapt this concept and consider small robots that are deployed from trucks and move on sidewalks. Closely related problems are truck-and-trailer routing problems as in Derigs et al. (2013) or Meisel and Kopfer (2014), in which trucks are active and trailers are passive means of transportation.

2.4 Platooning

This section is dedicated to describing the concept of platooning and its use cases. First, we provide a motivation in Section 2.4.1 before describing the technological advancements up to now in Section 2.4.2. Finally, we review relevant planning problems concerning the use of platooning with a focus on logistics applications in Section 2.4.3.

2.4.1 Motivation

Platooning or convoying is seen as an opportunity to utilize automated driving capabilities if conditions enable it. Figure 2.5 shows a schematic example for a truck platoon on a highway (VDA, 2020). When the driver of the first vehicle takes over the lead function for a platoon, drivers behind him can activate their automated driving system. The following vehicles merely replicate the dynamic driving tasks of the lead vehicle, which facilitates their driving environment considerably. Cooperative adaptive cruise control ensures vehicle following and speed control (Shladover et al., 2015). Vehicle-to-vehicle communication technologies may be used to transfer the control signals among the vehicles in the platoon. Beyond that, vehicle-to-infrastructure communication via road-side units may provide additional information to vehicles in certain traffic situations (Barrachina et al., 2013).

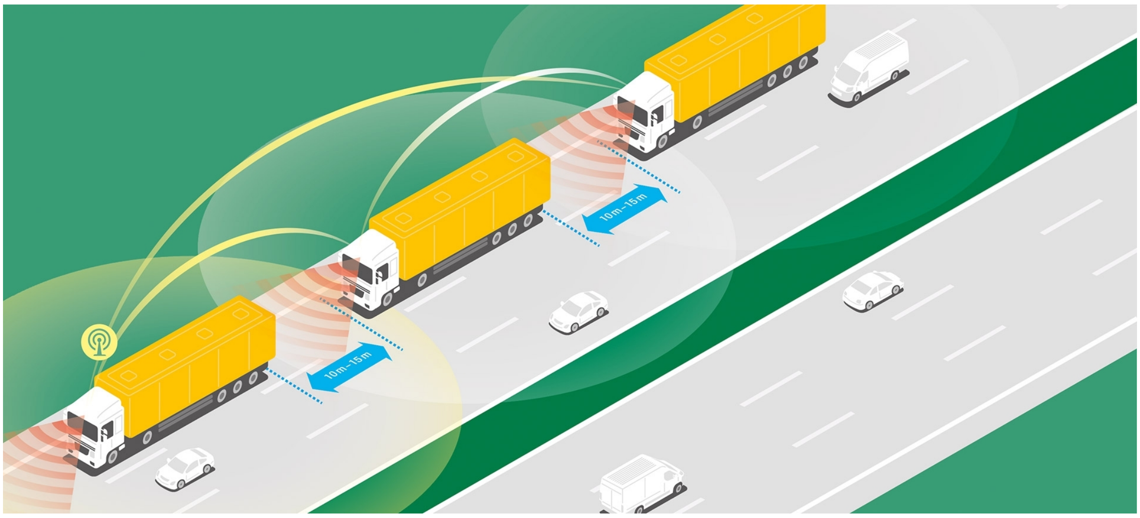


Figure 2.5: Schematic illustration of truck platooning (VDA, 2020).

There are several characteristics that motivate the adoption of platooning technology. First, it allows to automate the following vehicles as Schmeitz et al. (2019) show in a real-world use case in Eindhoven as part of the EU AUTOPILOT project. Thus, the workload on drivers and human errors can be reduced. Additional benefits of platooning include the occupation of less road space in contrast to individual vehicles requiring larger headway between each other (Lioris et al., 2016). Simulation results show that this can mitigate congestion and increase throughput (Wang et al., 2019). Additionally, experiments conducted by Lammert et al. (2014) show fuel savings of up to 9.7% for following vehicles in a platoon due to reduced aerodynamic drag. Experiments with a platoon of three fully-automated trucks by Tsugawa et al. (2011) show fuel savings of about 14%

with even the leader saving up to 7.5% due to reduced turbulence. Although automation can decrease the risk of collisions, safety issues may still arise due to possible cut-ins from surrounding traffic (Axelsson, 2017).

2.4.2 Technological advancements

The concept of platooning has existed for several decades in which the technology has been refined. In particular, there have been several large-scale projects that focus on platooning research and its real-world applications (Bergenhem et al., 2012). The PATH project was founded in 1986 by the California Department of Transportation and considers platoons on highways in which all vehicles are automated, including the leading vehicle (Shladover, 2007). After early experiments with passenger cars, more recent research focuses on heavy trucks. SARTRE is funded by the European Commission and considers platoons with manually driven lead trucks that are followed by automated trucks and passenger cars (Chan, 2016). Its research focuses on exploring vehicle-to-vehicle technology to ensure safe platooning without infrastructure changes. Energy ITS was initiated in Japan in 2008 to mitigate the lack of skilled drivers and to save energy (Tsugawa et al., 2011). A platoon of three fully-automated trucks is tested under not only steady state driving but also lane changing. The Grand Cooperative Driving Challenge was held in 2011 and 2016 to explore different platooning scenarios in both urban and highway environments (Englund et al., 2016). The participating teams perform cooperative platoon operations such as merging and splitting with a mix of passenger cars and trucks.

Nowadays, most automotive companies are engaged in the development of platooning technology. The European Automobile Manufacturers Association (2017) provides a roadmap with the goal of enabling “to drive across Europe on motorways (thus crossing national borders) with multi-brand platoons, without needing any specific exemptions” by 2023. The proposed steps towards this goal include initiatives, such as the European Truck Platooning Challenge, to foster research and development as well as to realize regulatory changes. A future goal of this EU roadmap is to pave the way for platooning with fully autonomous trucks. In the United States, the startup Peloton Technology offers third-party hardware and software kits that enable platooning among two trucks (McLane, 2019).

2.4.3 Planning problems

Utilizing the benefits of platooning raises a variety of planning issues for service providers. A comprehensive literature review that focuses on truck platooning on highways is given by Bhoopalam et al. (2018). Larsson et al. (2015) first introduce the vehicle platooning problem and formulate an integer program to derive fuel-optimal solutions for grouping trucks with an origin and a destination to form platoons. Due to the $\mathcal{NP}$-hardness of the problem, computational results are only provided for the so-called unlimited platooning problem which neglects delivery deadlines and restrictions on the platoon size. Larson et al. (2016) propose an improved formulation and are able to produce near-optimal solutions for larger instances of this problem. Boysen et al. (2018a) solve instances with time windows on networks where all trucks share an identical path. Larsen et al. (2019) propose a model for hub-based truck platooning and analyze different static and dynamic dispatching strategies.

While the previously introduced research focuses on the global grouping of trucks to platoons via some kind of centralized platform, other researchers consider local ad-hoc formations with decentralized approaches. Maiti et al. (2017) provide an ontological framework for ad-hoc platoon formations. Johansson and Martensson (2019) propose game-theoretic models to distribute profits among competing providers that collaborate in forming platoons. First ideas of using the advantages of platooning in an urban context are proposed by Agatz et al. (2016) with the related concept of flexible road trains. Sebe et al. (2019) group cross-provider platoons based on origin-destination pairs of autonomous pods.

2.5 Research gap

In conclusion of this chapter, we can state that there is a vast body of literature covering individual aspects of the four investigated topics but some links are missing between them. While tactical planning models exist for city logistics, the literature on integrating AVs and platoons is still scarce. This is partly due to fact that possible use cases for AVs and platoons, especially in urban environments, are still in experimental stages. Previous applications of automated driving and platooning mainly focus on highway traffic and, thus, long-haul transportation. However, to assess the impact these new technologies will have on the operations of city logistics service providers, tactical planning models may provide valuable insights.

Tactical planning provides a perspective of regularity and repeatability from which automated systems may benefit considerably. This is especially relevant to facilitate the interaction between humans and AVs that is necessary during commodity loading, transshipment, or delivery processes of an LSP. Platooning in particular requires an accurate coordination and synchronization of resources. Allocating the resources and scheduling their utilization in a tactical plan provides reliable guidelines for performing those operational processes on a regular basis. Also, the interaction of AVs and platoons with surrounding traffic in cities can be improved with tactical planning, e.g., by avoiding and mitigating regularly occurring congestion. From this, we conclude that SND problems should be enriched with the consideration of mixed autonomous fleets, that comprise AVs and utilize platooning, to support tactical planning for city logistics.

Chapter 3

Problem description

Based on the research gap identified in the previous chapter, we describe the service network design problem with mixed autonomous fleets (SNDMAF) in this chapter. The problem focuses on the tactical planning of the operation of autonomous vehicles and platoons in a city logistics environment. Thus, we formulate it as a service network design problem. In general, the SNDMAF considers an LSP that contributes to a city logistics network by transporting commodities. The LSP operates a mixed autonomous fleet that utilizes the underlying heterogeneous infrastructure by means of platooning. The objective is to determine a tactical plan which minimizes the total cost of the LSP in a planning horizon.

The problem consists of several aspects, which are explained in detail in the following sections. In Section 3.1, we describe the city logistics network. Then, we specify the features of the mixed autonomous fleet in Section 3.2 before describing the heterogeneous infrastructure in Section 3.3. In Section 3.4, we depict the characteristics of the commodities. We describe the tactical decisions of the LSP in Section 3.5 and conclude with defining the objective of the tactical plan in Section 3.6.

3.1 City logistics network

We consider a two-tier city logistics network that is used to transport commodities. The network comprises two different groups of facilities, external zones and satellites, that are used for handling commodities and loading vehicles.

An external zone denotes any kind of facility or area that connects a city or metropolitan area with long-distance transportation. Thus, it provides access to highways, ports, or airports and is typically located in the periphery of a city. Within an external zone, there typically exists at least one city distribution center (CDC), i.e., a facility in which commodities are sorted, stored, and loaded onto vehicles. CDCs can be operated solely by one LSP but also jointly with other transportation companies.

Satellites, on the other hand, are typically distributed within the city such that their locations cover large parts of the city. The satellite locations are chosen in a way to ensure both accessibility by trucks as well as close proximity to end customers. In contrast to external zones, satellites do not feature any warehousing capabilities for commodities. As they do, however, allow for the parking of vehicles, commodities can be transshipped between vehicles if they meet at a satellite at the same time.

The two tiers of city logistics are vertically connected to each other and satellites act as the physical interface between them. The first tier describes the transportation leg between external zones and satellites, for which trucks – also called "urban vehicles" (Crainic et al., 2009) – are used. In external zones, commodities are consolidated and loaded onto these trucks that then perform transportation services via which commodities are transported to satellites.

At satellites, the commodities are transshipped to smaller vehicles – so-called "city freighters" – that perform last-mile delivery routes visiting customers in the second tier. The vehicles are designed to cope with narrow roads in crowded cities as well as to reduce energy demand and emissions. Alongside vans, also (e-)bikes, robots, and drones can be used for deliveries in the second tier. Pick-up stations, that are located at or near satellites, offer an alternative in which customers picking up commodities at satellites may even replace deliveries in the second tier.

In the following, we focus on the first tier when producing a tactical plan with the SNDMAF. Since the locations of external zones do not change frequently and commodity volumes are aggregated, the environment in the first tier is relatively stable and, thus, suitable for tactical planning. In contrast to the first tier, the second tier is highly impacted by stochasticity due to varying customer orders and a considerable impact of traffic on travel times between customer visits. Thus, the second tier is not well suited for tactical planning but rather requires more dynamic decision-making in a subsequent operational planning problem. Nevertheless, the SNDMAF takes the aggregated commodity demand at satellites and the synchronization times with the second tier into account.

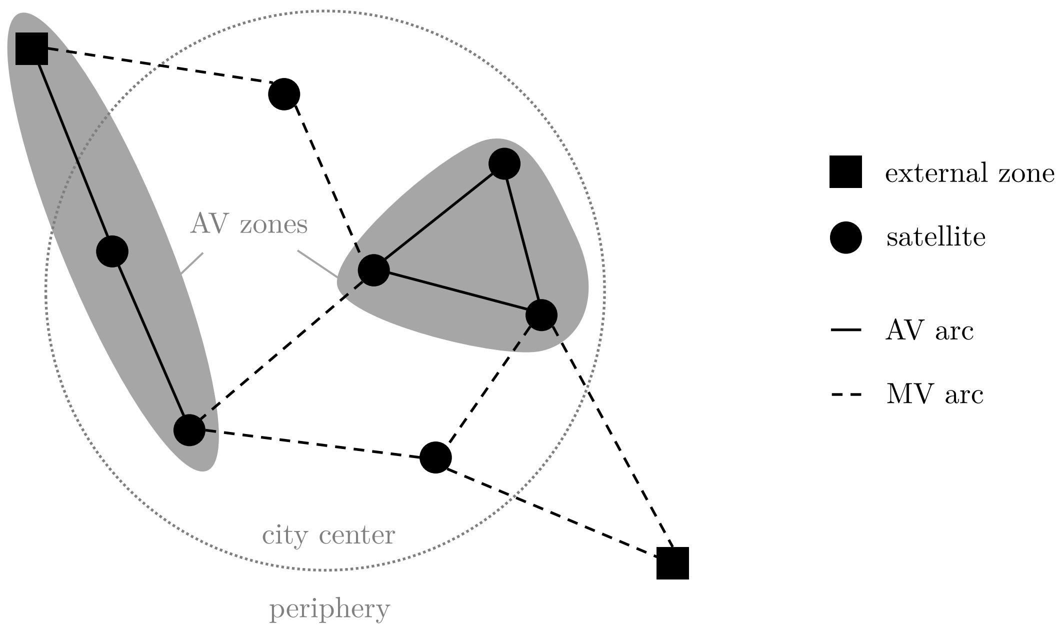


Figure 3.1: Problem setting for the first tier of city logistics with a mixed autonomous fleet, adapted from Scherr et al. (2019).

In Figure 3.1, we show an example for the first tier of a two-tier city logistics setting in which a mixed autonomous fleet is used. External zones are typically located at the periphery of the city and satellites are distributed within the city center. In the figure, we also show an example for a heterogeneous infrastructure with AV zones illustrated in a gray shade to denote clusters of AV arcs which differ from MV arcs. A more detailed description is given in the subsequent sections.

3.2 Mixed autonomous fleet

We assume that the LSP employs a mixed autonomous fleet for the first tier of city logistics. This fleet consists of two types of vehicles, autonomous vehicles (AVs) and manually operated vehicles (MVs). Both types feature a limited transportation capacity to load commodities and a certain driving speed at which the vehicles travel between the locations of the network. We assume that MVs require a human driver and are able to travel on any street of the city network. AVs, however, are only able to travel autonomously in specific parts of the network, which we denote as AV zones. Outside of these AV zones, they need to be pulled by MVs in platoons. The vehicles of a platoon are only linked by communication technologies and there is no need for physical coupling, such as in, e.g., truck-and-trailer configurations. One MV, however, can only pull a limited number of AVs at a time, which we denote as the platoon capacity, to limit the platoons to a feasible length within an urban environment.

Forming a platoon requires synchronization between MVs and AVs, as they need to meet at a location at the same time. We introduce the following three *platoon operations* that describe necessary activities to achieve synchronization:

- MVs and AVs can *merge* to form platoons. Also, AVs can *merge* with existing platoons consisting of multiple vehicles.
- On the contrary, a platoon can *split* by AVs leaving it. A platoon can dissolve completely if it only consisted of one MV and one AV or if an MV leaves a platoon.
- Finally, we consider that platoons *transit* through locations, i.e., the formation stays consistent between entering and leaving a location.

Figure 3.2 illustrates the platoon operations based on a small network segment. The nodes depict locations and the arcs depict road sequences between those locations. Starting from the left side of the figure, a platoon consisting of one MV and two AVs splits at the first location into a shorter platoon, consisting of one MV and one AV, and one AV traveling independently. The transit operation at the second location ensures that the formation of the platoon stays consistent between entering and leaving the location. At the third location, the short platoon and the AV merge to form a longer platoon again. Although we only show the spatial effect of the platoon operations in this example, note that synchronization can also be achieved by vehicles idling at a location. In this way, the respective merge or split operations can be depicted by additionally considering the temporal dimension.

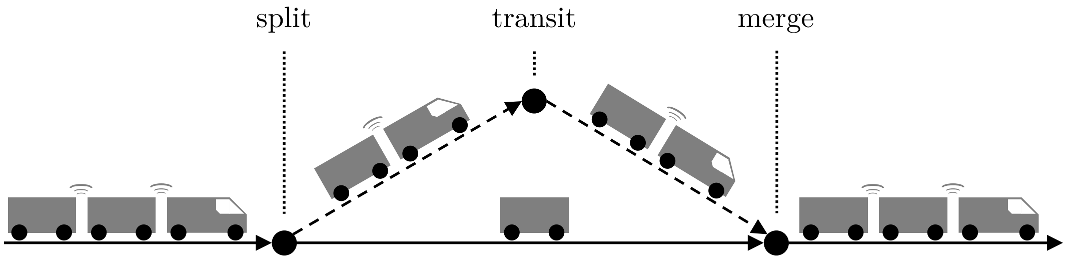


Figure 3.2: Platoon operations on a network segment.

Further note that AVs are not designated to one specific MV as a platoon leader but can switch between different platoons at a location by performing the necessary split and merge operations. Also, we do not explicitly determine the sequence of the individual vehicles in a platoon in this tactical problem. Instead, we consider that this sequence can be rearranged easily at an operational level. We further assume that performing platoon operations does not induce any significant effort in terms of cost and time as there is no need for physical coupling. Finally, we presume that the platoon operations are only performed at the locations that are depicted in our city logistics network, external zones and satellites. These locations provide the necessary space for maneuvering vehicles during merge and split operations, if required. Nevertheless, performing platoon operations at other locations in the network, e.g., relevant junctions at intermediate locations, may be possible.

As the tactical plan should be repeatable for every day in the medium-term time horizon, the vehicles should respect some further resource management requirements. First, each vehicle needs to start from one of the external zones at the beginning of the day and return to one of the external zones at the end of the day. Second, the number of MVs and the number of AVs need to be balanced between the beginning and end of the day at each external zone. The two requirements ensure repeatability while still allowing vehicles and drivers to return to a different external zone than they depart from. This assumption is widely applied in SND problems, e.g., in Pedersen et al. (2009) and Crainic et al. (2016), to provide some added flexibility to the LSP.

3.3 Heterogeneous infrastructure

The layout of AV zones that are scattered over the city forms a so-called heterogeneous infrastructure. As we consider a heterogeneous infrastructure for the city's street network, it also impacts our city logistics network. Usually, the LSP cannot directly decide upon the installment of AV zones which is rather within the responsibility of public authorities. Therefore, we consider a given heterogeneous infrastructure in this problem setting. However, different stakeholders, such as transportation companies among others, may impact or support the decisions on where to provide the necessary infrastructure adaptations to enable automated driving.

We distinguish between three types of *heterogeneous infrastructure*, that differ with respect to the layout of their AV zones, in the following.

- In the *corridor* infrastructure, several AV zones are separated from each other. This may be the case for industrial areas or parts of the network with low speed limits. On streets connecting the individual AV zones, AVs are not able to travel without guidance. Instead, they require to be platooned by an MV. In this setting, MVs may fulfill the primary task to transfer AVs between AV zones.

- In the *artery* infrastructure, AVs act as suppliers from external zones towards satellites in the city center using dedicated lanes on urban highways or arterial roads.

Existing infrastructure for public transport such as tram lines or bus lanes could be utilized for this purpose. Alternatively, there exist autonomous systems which use paint markings on the pavement for navigation (Stopard et al., 2015; Coren, 2017). In the *artery* infrastructure, AVs are only allowed to travel on a so-called artery that leads through the network. Using this path, AVs can supply MVs with commodities from the external zone. After being transshipped, the commodities can be delivered to satellites outside of this path by MVs.

- In the *nanny* infrastructure, the network for AVs is fragmented and the routes of MVs and AVs need to be closely related to each other. An MV may pull a number of AVs which decouple over short time periods whenever parallel delivery is demanded. This behavior is reminiscent to truck-and-drone routing, apart from widely different vehicle characteristics. In the *nanny* infrastructure, AVs can travel on specific parts of the network independently but require platooning in other parts.

We depict instances illustrating the three types of heterogeneous infrastructure in Figure 5.1, which we consider in the computational experiments presented in Chapter 5. While these three types may provide a classification for heterogeneous infrastructure, any combinations of them or configurations that may not fall into this classification may occur in the real world. With the street network of Braunschweig, Germany, we consider a real-world example later in this work. Since 2010, the ring road of this city is prepared for automated driving as part of a research study (Nothdurft et al., 2011). The resulting heterogeneous infrastructure combines aspects of the corridor and artery types. We investigate the deployment of a mixed autonomous fleet in this specific heterogeneous infrastructure in a case study presented in Chapter 8.

3.4 Commodities

Within the planning horizon, there is an expected demand for the commodities that need to be transported. Each commodity features a specific volume that is defined in discrete units. A pallet or a bag containing parcels could represent one such volume unit. Further, each commodity has an origin and a destination, both in terms of the location and the time point. For inbound demand, which typically represents the majority of the parcel volume in a city, the origin specifies the external zone where the commodity appears and the time after which it is available to be picked up. The destination denotes the satellite to which the commodity needs to be delivered and the time point of arrival.

In a traditional next-day delivery setting, the commodities are typically available at the beginning of the day, whereas in same-day delivery settings, commodities may arrive at the external zones during the day. Additionally, outbound demand can be considered in a city logistics setting, e.g., by modeling returns of commodities, to improve load utilization of vehicles. Outbound demand defines commodity origins at satellites and destinations at external zones.

Commodities can be transshipped between the vehicles of the first tier at any satellite to enhance consolidation and to avoid poorly loaded vehicles. Depending on the content and the packaging of the commodities, splitting them into discrete parts and delivering the parts over separate paths may be allowed.

As commodities cannot be stored at satellites, precise arrival times at satellites are predefined for the first tier of city logistics. These time points also determine the synchronization with the second tier and, thus, the transshipment of commodities between the vehicles of the two tiers. We assume that commodities are demanded to arrive at satellites regularly distributed over the day to balance the workload of the second tier for the following two reasons. First, the second-tier vehicles, such as small vans or cargo bikes, as well as pickup stations usually feature only limited load capacity. Second, waiting delivery persons or empty pickup stations are costly and not customer-friendly.

3.5 Tactical decisions

Tactical planning aims to determine a plan to utilize resources in the most efficient way over a medium-term time period (Crainic and Laporte, 1997). In this specific problem setting, the LSP considers demand forecasts over a medium-term time period, such as a month or a season. The tactical plan is then determined for a planning horizon, e.g., a day, such that it can be repeated over the medium-term time period without major changes. Tactical planning may neglect details of the implementation and therefore serves as an optimistic approximation of the operational planning level.

The LSP makes the following decisions within the tactical plan.

- The *fleet size* as well as the *fleet mix* between MVs and AVs are determined. With this decision, a number of MVs including their drivers and a number of AVs are allocated to each of the external zones.

- *MV* and *AV services* are scheduled for traveling between locations at specific time points. The services are required to be designed in a way that feasible routes can be derived for each individual vehicle. For services that involve platooning, the necessary platoon operations can also be derived.

- The *commodity routing* from their origin to their destination is determined by coupling commodities to services while respecting the available transportation capacity.

- Alternatively, *outsourcing* the delivery of a set of commodities to a third-party service provider can be chosen. With this decision, the LSP does not need to deliver the respective commodity. However, the cost for outsourcing is typically high.

3.6 Objective of the tactical plan

The objective of the LSP is to minimize the total costs associated with fulfilling customer demand. Although fleet acquisition costs and wages are usually paid longer-term, we aim to consider them jointly with the rather short-term costs for transportation. Thus, we assume the fixed costs are broken down according to the length of the planning horizon. As MVs require a human driver, we assume they would typically cause higher fixed costs than AVs. Finally, the total costs of the LSP are composed of the following cost types.

- *Fixed costs* are caused for assigning a vehicle of a specific type to the fleet. They translate to the broken down costs for acquiring and maintaining a vehicle as well as for driver salaries.
- *Service costs* denote the operational expenses for routing vehicles between locations, accounting for their energy consumption and wear. In the considered city logistics setting associated with low speeds, the effect of platooning on energy consumption and, thus, on service costs is negligible.
- *Transportation costs* apply for transporting a volume of commodities on vehicles, thus representing expenses for additional load and handling effort.
- *Outsourcing costs* need to be paid for contracting a third-party service provider to deliver a commodity if the commodity is not delivered by the LSP.

The SNDMAF, being a tactical planning problem, does not provide a solution that can then be directly implemented at the operational level. However, it provides guidelines for a subsequent operational planning problem, in which the specific vehicles and drivers are assigned to conduct services or to execute platoon operations. An LSP also typically decides on the loading and transshipment of commodities onto the planned services based on the actual demand that is realized shortly ahead of the given day. Thus, enlarging the scope of the tactical planning problem by integrating excessive operational issues can be counterproductive. An overly precise tactical plan would require significant adjustments for the slightest deviation from the demand forecast. Also, the resulting problem can quickly become too complex which may prevent applying it in practice. To this end, we refrain from integrating further operational considerations in the SNDMAF and instead consider a subsequent problem concerned with the implementation of a tactical plan at the operational level in Chapter 7.

Nevertheless, the tactical plan can also inform other decision-making processes, despite not considering them explicitly in the planning problem. Particularly for the second tier of city logistics, it provides reliability in terms of the arrival times of vehicles and expected commodity volumes that need to be handled at satellites. At facilities, workers may be scheduled, e.g., based on the arrival times of vehicles determined in the tactical plan. Outside of transportation, aggregated information from the plan can also be used in sales and marketing for quoting transportation capacities to potential customers.

In this chapter, we introduced and described the SNDMAF. The subsequent two chapters build upon this problem description. In Chapter 4, we provide a mathematical model for the SNDMAF using a time-expanded network formulation. We further describe model modifications for depicting operational policies an LSP may consider regarding the management of resources and the handling of commodities. In Chapter 5, we present computational experiments to validate the model and to study the impact of different problem characteristics on the structure of the solutions.

Chapter 4

Model

In this chapter, we provide a model for the service network design problem with mixed autonomous fleets (SNDMAF). We define the underlying time-expanded network and the notation of the model in Section 4.1. In Section 4.2, we formulate the mathematical model for the SNDMAF in its general form. Finally, we describe modifications to the model in Section 4.3.

4.1 Notation

In this section, we formally introduce the notation of the SNDMAF model. To explicitly consider the time-dependent properties of the problem, a time-expanded network is used to define the notation. We first describe the underlying physical network in Section 4.1.1 before describing the time-expanded network in Section 4.1.2. Finally, we state the variables and parameters used in the model in Section 4.1.3.

4.1.1 Physical network

We now introduce the notation of the underlying physical network $D_{ph} = (N_{ph}, A_{ph})$, also called the "flat" network. Table 4.1 provides an overview of its notation. The directed network D_{ph} includes the set of locations that are considered by the LSP and the road sequences connecting them. The set of nodes $N_{ph} = \{i\}$ contains the locations. A location i is either in the subset $N_E \subset N_{ph}$ if it denotes an external zone or in the subset $N_S \subset N_{ph}$ if it denotes a satellite. A location cannot belong to both subsets, i.e., $N_E \cap N_S = \emptyset$. The set A_{ph} contains physical arcs (i, j) connecting the locations i and j. The non-negative weight on the physical arcs is denoted as the travel time τ_{ij} between the locations.

While MVs can travel on all physical arcs without restrictions, AVs can only travel on a predefined subset of physical arcs. Therefore, we distinguish between AV arcs, on which AVs can travel without restrictions, and MV arcs, on which AVs can only travel if they are platooned by an MV. A physical arc is either an MV arc in subset $A_M \subset A_{ph}$ or an AV arc in subset $A_A \subset A_{ph}$, with $A_M \cap A_A = \emptyset$. Physical arcs with $i = j$ belong to the set of AV arcs A_A to allow AVs to idle at any location.

Table 4.1: Notation of the physical network D_{ph}, adapted from Scherr et al. (2020).

Notation	Description
$i \in N_{ph}$	Location
$i \in N_E$	External zone
$i \in N_S$	Satellite
$(i, j) \in A_{ph}$	Physical arc
$(i, j) \in A_A$	AV arc
$(i, j) \in A_M$	MV arc
$\tau_{ij} \in \mathbb{R}^{\geq 0}$	Travel time on physical arc $(i, j) \in A_{ph}$

4.1.2 Time-expanded network

The time-expanded network $D = (N, A)$, also a directed network, is created based on the physical network. We describe its notation in the following and provide an overview in Table 4.2. The locations i are replicated in discrete time periods $t \in T = \{0, \Delta, \ldots, t_{max}\}$ with t_{max} being the length of the planning horizon. We define the discretization $\Delta \in \mathbb{N}$ of the time-expanded network as the length of the intervals between subsequent time periods. By replicating every location once in every time period, we obtain the set N with nodes (i, t), that are described both in the spatial and temporal dimension. An arc $((i, t), (j, \bar{t}))$

in the time-expanded network connects the node (i,t) with the node $(j,\bar{t})$.

To depict vehicle-related decisions with an adequate level of detail, the set of arcs A can be distinguished into auxiliary, holding, waiting, and movement arcs.

- *Auxiliary arcs* in $A_\gamma \subset A$ provide a way to model the fleet assignment and the requirement for cyclic vehicle routes. They connect the last (t_{max}) and the first (t_0) replication of an external zone, i.e., $i = j$ with $i,j \in N_E$, by going back in time.

- *Holding arcs* in $A_h \subset A$ connect nodes of the same external zone, i.e., $i = j$ with $i,j \in N_E$, in subsequent time periods to model idling vehicles and the storage of commodities.

- *Waiting arcs* in $A_w \subset A$ connect nodes of the same satellite, i.e., $i = j$ with $i,j \in N_S$. In contrast to holding arcs, they only allow for idling of vehicles but not for the storage of commodities.

- *Movement arcs* in $A_m \subset A$ encode traveling between two locations, i.e., $i \neq j$ with $i,j \in N_E \cup N_S$. The length of these arcs, $\bar{t} - t$, is determined by the respective travel time τ_{ij} obtained from the physical network.

Table 4.2: Notation of the time-expanded network D, adapted from Scherr et al. (2020).

Notation	Description
$t \in T = \{0, \Delta, \dots, t_{max}\}$	Time period in planning horizon t_{max}
$\Delta \in \mathbb{N}$	Length of time intervals (discretization)
$(i,t) \in N$	Node
$((i,t),(j,\bar{t})) \in A$	Arc
$((i,t),(j,\bar{t})) \in A_\gamma$	Auxiliary arc to assign number of vehicles
$((i,t),(j,\bar{t})) \in A_h$	Holding arc to idle at external zone
$((i,t),(j,\bar{t})) \in A_w$	Waiting arc to idle at satellite
$((i,t),(j,\bar{t})) \in A_m$	Movement arc to travel between locations

Figure 4.1 illustrates the process of transforming a small physical network (left figure) with three locations to a corresponding time-expanded network (right figure) based on the previous description. The locations of the physical network, one external zone and

two satellites, are depicted on the y-axis of the time-expanded network. The temporal dimension depicted on the x-axis of the time-expanded network features a discretization of $\Delta = 10$ minutes. Thus, the time-expanded network contains the time periods $t \in T = \{0, 10, \ldots, t_{max}\}$. Note that the time periods between $t = 20$ and t_{max} are not depicted in this example for the sake of simplicity. The discretization allows to accurately translate the travel times of the arcs in the physical network, i.e., $\tau_{01} = \tau_{10} = 20$ minutes and $\tau_{12} = \tau_{21} = 10$ minutes, to the length of the movement arcs in the time-expanded network. We distinguish between the different types of arcs in the networks by illustrating them in different line styles. The physical network contains MV and AV arcs, whereas the time-expanded network contains auxiliary, holding, waiting, and movement arcs.

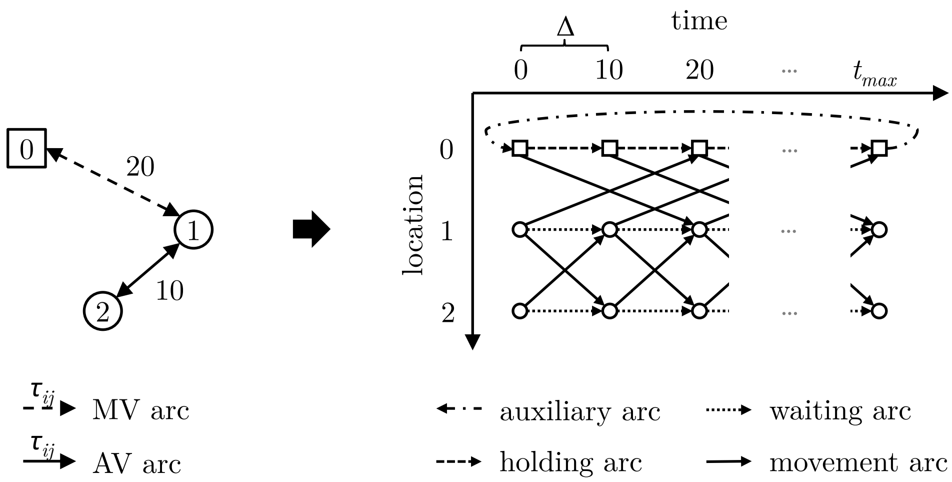


Figure 4.1: Transforming a physical network to a time-expanded network, adapted from Scherr et al. (2020).

4.1.3 Variables and parameters

Before presenting the variables, we formally define the set K that contains the commodities. Each commodity $k \in K$ is characterized by a volume $v^k \in \mathbb{N}$ in discrete units, an origin node $o^k = (i_o^k, t_o^k)$, and a destination node $d^k = (i_d^k, t_d^k)$. An origin comprises an origin location i_o^k and a departure time t_o^k. A destination comprises a destination location i_d^k and an arrival time t_d^k. The storage capacity to hold commodities at an external zone is limited to the parameter u_E.

The vehicles of the LSP are divided into two types, MVs and AVs. Both types of vehicles are able to perform services to transport commodities. The transportation capacity of an MV is limited to the parameter u_M, the transportation capacity of an AV is limited to u_A. MVs can travel on all arcs in set A of the time-expanded network. AVs, however, can only move independently on arcs of the time-expanded network which depict traveling on an AV arc, i.e., $((i,t),(j,\bar{t})) \in A : (i,j) \in A_A \subset A_{ph}$. This set includes all holding and waiting arcs such that idling of AVs is permitted at any location. On movement arcs for which the physical arc belongs to the set of MV arcs, i.e., $((i,t),(j,\bar{t})) \in A_m : (i,j) \in A_M \subset A_{ph}$, AVs can only travel if they are platooned by an MV. The platoon capacity n_{ij}^P defines the maximum number of AVs that can follow one MV in a platoon and is specified for each physical arc $(i,j) \in A_{ph}$.

Based on the time-expanded network $D = (N, A)$ and the set of commodities K, the variables of the model are defined as follows.

- *Service variables* for MVs, $m_{ij}^{t\bar{t}} \in \mathbb{N}^0$, and for AVs, $a_{ij}^{t\bar{t}} \in \mathbb{N}^0$, are defined as integer variables. Their value denotes the number of transportation services that are installed on an arc $((i,t),(j,\bar{t})) \in A$. The number of services on an auxiliary arc (A_γ) indicates the number of MVs or AVs that are allocated to the respective external zone and that are available over the complete planning horizon.

- *Commodity flow variables* $x_{ij}^{kt\bar{t}} \in \{0,1\}$ equal one if a commodity k is transported on an arc $((i,t),(j,\bar{t})) \in A$, otherwise zero. Generally, the variables are defined as binary variables such that splitting a commodity during its transportation from origin to destination is not permitted.

- *Outsourcing variables* $y^k \in \{0,1\}$ are defined as binary variables. If the value is one, the delivery of commodity k is outsourced to a third-party service provider, if the value is zero, the delivery has to be fulfilled by the fleet of the LSP.

Finally, we distinguish between the following types of costs that are activated by the variables.

- *Fixed costs* $f_M \in \mathbb{R}^+$ for MVs or $f_A \in \mathbb{R}^+$ for AVs are caused for assigning a vehicle of a specific type to the fleet. They are applied to the service variables on the auxiliary arcs of the time-expanded network.
- *Service costs* $g_{ij} \in \mathbb{R}^+$ for MVs and $l_{ij} \in \mathbb{R}^+$ for AVs denote the costs for a vehicle traveling on a movement arc using the service variables. The costs are specified for each physical arc $(i, j) \in A_{ph}$ such that they do not depend on the time period.
- *Transportation costs* $c_{ij}^k \in \mathbb{R}^+$ apply for transporting a commodity k on a physical arc $(i, j) \in A_{ph}$ using the commodity flow variables.
- *Outsourcing costs* $b^k \in \mathbb{R}^+$ are linked to the use of the outsourcing variable for a commodity k.

We give on overview of the variables in Table 4.3 and of the parameters in Table 4.4.

Table 4.3: Notation of the variables, adapted from Scherr et al. (2020).

Notation	Description
$m_{ij}^{t\bar{t}} \in \mathbb{N}^0$	MV services on arc $((i, t), (j, \bar{t})) \in A$
$a_{ij}^{t\bar{t}} \in \mathbb{N}^0$	AV services on arc $((i, t), (j, \bar{t})) \in A$
$x_{ij}^{kt\bar{t}} \in \{0, 1\}$	Flow of commodities $k \in K$ on arc $((i, t), (j, \bar{t})) \in A$
$y^k \in \{0, 1\}$	Third-party service for commodity $k \in K$

Table 4.4: Notation of the parameters, adapted from Scherr et al. (2020).

Notation	Description
$k \in K$	Commodities
$v^k \in \mathbb{N}$	Volume in units of commodity $k \in K$
$o^k = (i_o^k, t_o^k) \in N$	Origin location and departure time of commodity $k \in K$
$d^k = (i_d^k, t_d^k) \in N$	Destination location and arrival time of commodity $k \in K$
$u_E \in \mathbb{N}$	Warehousing capacity of external zones in volume units
$u_M \in \mathbb{N}$	Transportation capacity of MVs in volume units
$u_A \in \mathbb{N}$	Transportation capacity of AVs in volume units
$n_{ij}^P \in \mathbb{N}$	Platoon capacity on a physical arc (max. number of AVs in a platoon)
$n_M \in \mathbb{N}$	Maximum number of available MVs
$n_A \in \mathbb{N}$	Maximum number of available AVs
$f_M \in \mathbb{R}^+$	Fixed cost for assigning an MV
$f_A \in \mathbb{R}^+$	Fixed cost for assigning an AV
$g_{ij} \in \mathbb{R}^+$	Service cost per MV service on arc $(i, j) \in A_{ph}$
$l_{ij} \in \mathbb{R}^+$	Service cost per AV service on arc $(i, j) \in A_{ph}$
$c_{ij}^k \in \mathbb{R}^+$	Transportation cost for commodity k on arc $(i, j) \in A_{ph}$
$b^k \in \mathbb{R}^+$	Cost for outsourcing delivery of commodity k to third-party service

4.2 Mathematical formulation

In this section, we formulate an integer program for the SNDMAF. The model we seek to solve is as follows:

$$\begin{aligned}\min \sum_{((i,t),(j,\bar{t}))\in A_\gamma} \left[f_M m_{ij}^{t\bar{t}} + f_A a_{ij}^{t\bar{t}}\right] + \sum_{((i,t),(j,\bar{t}))\in A_m} \left[g_{ij} m_{ij}^{t\bar{t}} + l_{ij} a_{ij}^{t\bar{t}}\right] \\ + \sum_{k\in K}\left[\sum_{((i,t),(j,\bar{t}))\in A_m} c_{ij}^k x_{ij}^{kt\bar{t}} + b^k y^k\right]\end{aligned} \tag{4.1}$$

subject to

$$\sum_{((j,\bar{t}),(i,t))\in A} x_{ji}^{k\bar{t}t} - \sum_{((i,t),(j,\bar{t}))\in A} x_{ij}^{kt\bar{t}} = \begin{cases} -1 + y^k, & (i,t) = o^k, \\ 1 - y^k, & (i,t) = d^k, \\ 0, & (i,t) \neq o^k, d^k, \end{cases} \quad \forall k \in K, (i,t) \in N \tag{4.2}$$

$$\sum_{k \in K} x_{ij}^{kt\bar{t}} v^k \leq u_E, \qquad \forall((i,t),(j,\bar{t})) \in A_h \tag{4.3}$$

$$\sum_{k \in K} x_{ij}^{kt\bar{t}} v^k \leq u_M m_{ij}^{t\bar{t}} + u_A a_{ij}^{t\bar{t}}, \qquad \forall((i,t),(j,\bar{t})) \in A \setminus A_h \tag{4.4}$$

$$a_{ij}^{t\bar{t}} \leq n_{ij}^P m_{ij}^{t\bar{t}}, \qquad \forall((i,t),(j,\bar{t})) \in A_m : (i,j) \in A_M \tag{4.5}$$

$$\sum_{((j,\bar{t}),(i,t)) \in A} m_{ji}^{\bar{t}t} = \sum_{((i,t),(j,\bar{t})) \in A} m_{ij}^{t\bar{t}}, \qquad \forall(i,t) \in N \tag{4.6}$$

$$\sum_{((j,\bar{t}),(i,t)) \in A} a_{ji}^{\bar{t}t} = \sum_{((i,t),(j,\bar{t})) \in A} a_{ij}^{t\bar{t}}, \qquad \forall(i,t) \in N \tag{4.7}$$

$$x_{ij}^{kt\bar{t}} \in \{0,1\}, \qquad \forall k \in K, ((i,t),(j,\bar{t})) \in A \tag{4.8}$$

$$y^k \in \{0,1\}, \qquad \forall k \in K \tag{4.9}$$

$$m_{ij}^{t\bar{t}},\ a_{ij}^{t\bar{t}} \in \mathbb{N}^0, \qquad \forall((i,t),(j,\bar{t})) \in A \tag{4.10}$$

The objective function (4.1) minimizes the total costs of the LSP. The first term comprises the fixed costs for assigning the fleet of MVs and AVs. The second term contains the service costs for performing MV and AV transportation services. The third term includes the costs for the transportation of commodities and the outsourcing costs in case the delivery is conducted by a third-party service provider.

The model is subject to the following constraints. Constraints (4.2) depict the flow conservation for the commodities. They are formulated for three cases in which a node is either defined as the origin, the destination, or an intermediate stop. If the value of the commodity's corresponding outsourcing variable is set to one, the first two cases are overridden and the flow conservation constraints are rendered unnecessary.

Constraints (4.3) and (4.4) denote the capacity restriction for the commodities. Constraints (4.3) limit the storage of commodities to the warehousing capacity u_E of an external zone and are applied to holding arcs. Constraints (4.4) limit the transportation load to the capacity of MVs (u_M) and AVs (u_A) on all arcs except for holding arcs, i.e., on all arcs in set $A \setminus A_h$.

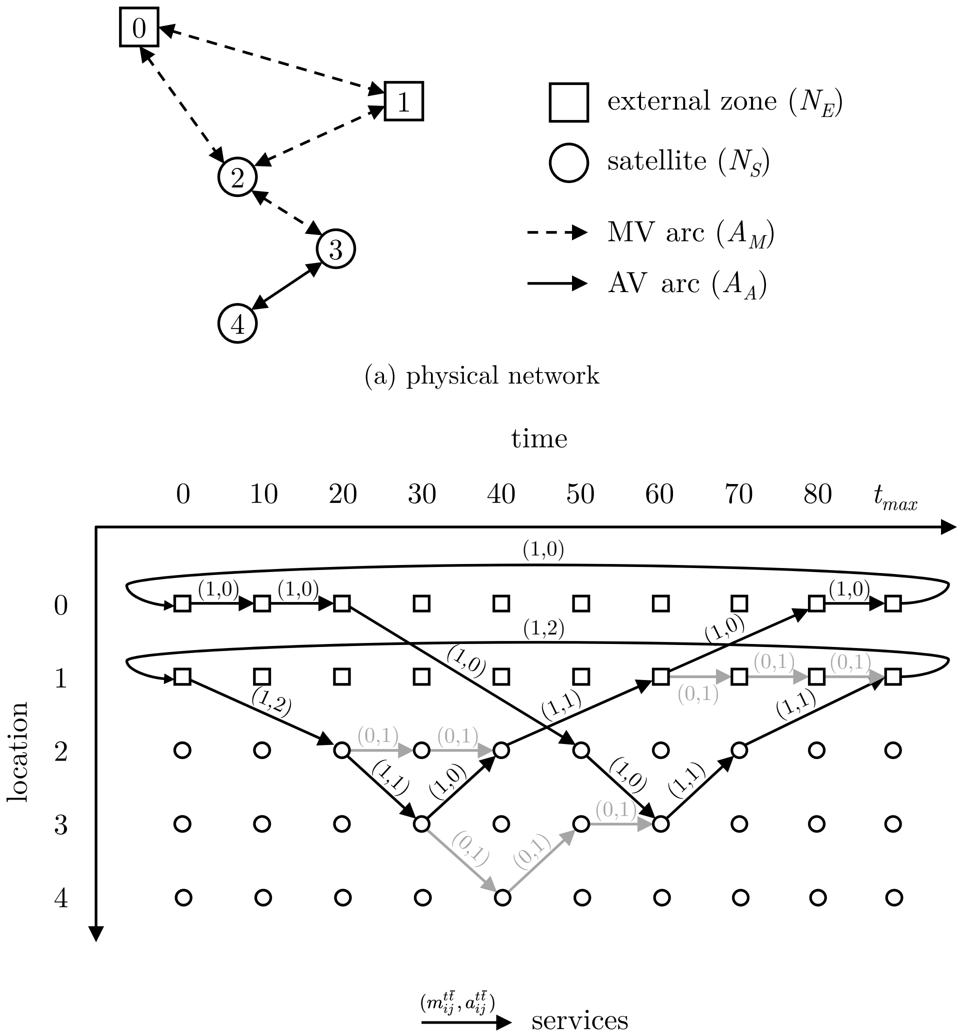
We denote Constraints (4.5) as the platooning constraints that link AV services to MV services on all movement arcs corresponding to MV arcs in the physical network. The parameter n_{ij}^{P} denotes the platoon capacity of an MV and limits the number of AVs in each platoon.

Constraints (4.6) and (4.7) are the design-balance constraints that are applied to the service variables and basically resemble flow conservation (Pedersen et al., 2009). By balancing the services at each node, they enable the design of feasible routes. Constraints (4.6) ensure design balance for MV services, while Constraints (4.7) do so for AV services.

Finally, the domains of the variables are defined. Commodity flow and outsourcing variables are defined to be binary in Constraints (4.8) and (4.9), MV and AV service variables are defined to be integer in Constraints (4.10).

In Figure 4.2, we illustrate an example for the MV and AV services in an SNDMAF solution. Therefore, we first define an underlying physical network in Figure 4.2a representing two external zones and three satellites in a simple layout. Except for the AV arcs connecting satellites 3 and 4, all arcs are MV arcs. From this physical network, we derive the time-expanded network for the time periods $t \in T = \{0, 10, \ldots, t_{max}\}$, which is depicted in Figure 4.2b. The figure shows all nodes but only the arcs on which there is at least one service installed. The arcs are labeled with a tuple $(m_{ij}^{t\bar{t}}, a_{ij}^{t\bar{t}})$ indicating the number of MV services and AV services installed on them, i.e., the respective values of the service variables $m_{ij}^{t\bar{t}}$ and $a_{ij}^{t\bar{t}}$ in the SNDMAF solution. Further, arcs with at least one MV service ($m_{ij}^{t\bar{t}} > 0$) are shaded in black, whereas arcs with AV services only ($m_{ij}^{t\bar{t}} = 0$ and $a_{ij}^{t\bar{t}} > 0$) are shaded in gray.

In the illustrated solution, two MVs and two AVs are assigned to the fleet as can be read from the services installed on the auxiliary arcs. While ensuring that the number of vehicles per type is balanced at each external zone between the beginning and end of the planning horizon, each of the two MVs returns to a different external zone than it starts from. As design balance is ensured at nodes, MV and AV services also form feasible routes over the planning horizon. Platooning can be observed in the solution on those arcs on which both MV and AV services are installed, e.g., on arc $((1, 0), (2, 20))$. From



(a) physical network

(b) time-expanded network

Figure 4.2: Services in an exemplary SNDMAF solution, adapted from Scherr et al. (2019).

the two AVs traveling on this arc, one AV splits from the platoon at node $(2, 20)$ to wait at the respective satellite before re-joining the platoon at node $(2, 40)$. This AV could be used to extend the transportation capacity of the respective MV leading the platoon. The other AV travels independently on the AV arcs connecting the satellites 3 and 4 between splitting from the platoon and joining another platoon with a different leading MV at a later time point.

Since SND formulations such as the SNDMAF contain multiple combinatorial elements, they typically yield relatively weak linear programming (LP) relaxations. Thus, we provide two types of *valid inequalities* that strengthen the LP relaxation to improve the solvability of the model. In the following, we first consider disaggregate coupling constraints in Section 4.2.1 and then tighten the capacity constraints in Section 4.2.2.

4.2.1 Disaggregate coupling constraints

The capacity constraints (4.4) in the SNDMAF model couple the commodity flow on an arc to the respective services on that arc. In this way, commodities can only flow on an arc if the installed services provide enough capacity. Conversely, commodity flow is "forced" to be zero if there exists no service in the design. Crainic et al. (2001) entitle these constraints the weak forcing constraints as they lead to weak lower bounds obtained through LP relaxations. The authors propose the additional use of a redundant set of constraints, the so-called strong forcing constraints, to improve the LP relaxation.

In the following, we present constraints that resemble those strong forcing constraints. These constraints consider each commodity in a disaggregated form rather than in an aggregated form as in the weak forcing constraints. In our model, the disaggregate coupling constraints (4.11) and (4.12) enforce that there needs to be a service installed on each arc with a positive commodity flow, disregarding the capacities. The constraints are applied to movement and waiting arcs only, as commodities only need to be coupled with vehicles on these types of arcs.

We further distinguish between two types of disaggregate coupling constraints, depending on the type of arc in the physical network, i.e., MV or AV arc, they are applied to. We apply Constraints (4.11) to all waiting arcs as well as to movement arcs on AV arcs. On these types of arcs, both MVs and AVs could travel alone. The constraints ensure that a commodity only flows on the arc if at least one service of either type is installed on it. On movement arcs that are MV arcs, however, AVs can only travel in platoons which are always led by an MV. Thus, it is sufficient to only consider MV services in the disaggregate coupling constraints (4.12) on MV arcs.

$$x_{ij}^{kt\bar{t}} \leq m_{ij}^{t\bar{t}} + a_{ij}^{t\bar{t}}, \qquad \forall k \in K, ((i,t),(j,\bar{t})) \in A_w \cup A_m : (i,j) \in A_A \tag{4.11}$$

$$x_{ij}^{kt\bar{t}} \leq m_{ij}^{t\bar{t}}, \qquad \forall k \in K, ((i,t),(j,\bar{t})) \in A_m : (i,j) \in A_M \tag{4.12}$$

4.2.2 Commodity volume aggregation

We now aim to further tighten the capacity constraints (4.3) for storage and (4.4) for transportation. Thus, we optionally substitute them with the constraints (4.13) and (4.14). For this reason, we determine the total possible commodity volume $V_{ij}^{t\bar{t}}$ for every arc in the time-expanded network. The volume $V_{ij}^{t\bar{t}}$ is defined as the aggregated volume of all commodities that can feasibly flow on the arc. Feasibility in this case is given if a commodity is able to flow on an arc based on a time-dependent path between the earliest departure time at the origin and the latest arrival time at the destination.

$$\sum_{k \in K} x_{ij}^{kt\bar{t}} v^k \leq \min\{u_E, V_{ij}^{t\bar{t}}\}, \qquad \forall ((i,t),(j,\bar{t})) \in A_h \tag{4.13}$$

$$\sum_{k \in K} x_{ij}^{kt\bar{t}} v^k \leq \min\{u_M, V_{ij}^{t\bar{t}}\} m_{ij}^{t\bar{t}} + \min\{u_A, V_{ij}^{t\bar{t}}\} a_{ij}^{t\bar{t}}, \qquad \forall ((i,t),(j,\bar{t})) \in A \setminus A_h \tag{4.14}$$

4.3 Modifications

In this section, we introduce three optional modifications of the generic SNDMAF model we have stated in (4.1)–(4.10). These modifications are concerned with particular operational policies that the LSP might want to impose given the specific problem setting. We consider the splitting of commodities (Section 4.3.1), additional fleet management considerations (Section 4.3.2), and a fixed external zone assignment for MVs (Section 4.3.3).

4.3.1 Commodity flow splitting

In the model as previously defined, a commodity k is not allowed to be split and, thus, the flow variables $x_{ij}^{kt\bar{t}}$ are denoted as binary. Simply denoting the flow variables as continuous may provide infeasible solutions as commodities cannot be split into arbitrary parts. It may, however, be useful for an LSP to split commodities into discrete parts and deliver them over separate paths, e.g., if these commodities are aggregated in the first place or can be handled with little effort.

To model this case, we provide an alternative version of the commodity flow conservation constraints (4.2) in the following. Constraints (4.15) ensure that the flow at an origin or destination of a commodity equals the total volume of the commodity. By considering integer commodity flow variables as defined in (4.16) instead of the binary definition in (4.8), commodities can be broken into discrete parts. Note that the respective transformation from binary to integer variables can also be performed for the outsourcing variables y^k. In this case, those variables would enable outsourcing the delivery of parts of a commodity.

$$\sum_{((j,\bar{t}),(i,t))\in A} x_{ji}^{k\bar{t}t} - \sum_{((i,t),(j,\bar{t}))\in A} x_{ij}^{kt\bar{t}} = \begin{cases} v^k(-1+y^k), & (i,t) = o^k, \\ v^k(1-y^k), & (i,t) = d^k, \\ 0, & (i,t) \neq o^k, d^k, \end{cases} \quad \forall k \in K, (i,t) \in N \tag{4.15}$$

$$x_{ij}^{kt\bar{t}} \in \mathbb{N}^0, \quad \forall k \in K, ((i,t),(j,\bar{t})) \in A \tag{4.16}$$

4.3.2 Fleet management considerations

Decisions about the fleet size and mix can be constrained in practice, e.g., by an LSP's ability to acquire vehicles or hire drivers. Thus, we provide an optional set of constraints to model enhanced fleet management considerations. Similar to the resource bound constraints introduced by Crainic et al. (2016), these constraints limit the maximum number of vehicles per type the LSP can assign. Therefore, we define the parameter n_M to denote a maximum number of MVs and the parameter n_A to denote a maximum number of AVs. As optional fleet size constraints, Constraints (4.17) restrict the number of MVs to parameter n_M and Constraints (4.18) restrict the number of AVs to n_A, respectively. The restriction is imposed on all auxiliary arcs connecting the last replication with the first replication of an external zone.

$$\sum_{((i,t),(j,\bar{t}))\in A_\gamma} m_{ij}^{t\bar{t}} \leq n_M, \tag{4.17}$$

$$\sum_{((i,t),(j,\bar{t}))\in A_\gamma} a_{ij}^{t\bar{t}} \leq n_A, \tag{4.18}$$

4.3.3 Fixed external zone assignment of MVs

An LSP may want to enforce particular policies that require enhanced control over the individual vehicles of the fleet. In this section, we present an alternative model formulation which particularly enforces each MV to return to the same external zone it departs from. For this alternative model formulation, we modify the generic model formulated by Equations (4.1)–(4.10).

First, we introduce the set of MVs $P = \{p\}$ to contain an index p for each individual MV. Based on this, the binary variables $m_{ijp}^{t\bar{t}} \in \{0,1\}$ denote whether the MV p performs a service on arc $((i,t),(j,\bar{t})) \in A$. Since we have to determine a magnitude $|P|$, the size of set P restricts the fleet size similar to the fleet management constraints defined in (4.17). The same procedure can be applied to create a set of individual AVs to control AV services more explicitly. The alternative model formulation respecting the set of MVs P uses the objective function (4.19) that is adapted from the objective function (4.1) used in the generic SNDMAF model.

$$\begin{aligned} \min \sum_{((i,t),(j,\bar{t}))\in A_\gamma} \left[\sum_{p\in P} f_M m_{ijp}^{t\bar{t}} + f_A a_{ij}^{t\bar{t}}\right] + \sum_{((i,t),(j,\bar{t}))\in A_m} \left[\sum_{p\in P} g_{ij} m_{ijp}^{t\bar{t}} + l_{ij} a_{ij}^{t\bar{t}}\right] \\ + \sum_{k\in K} \left[\sum_{((i,t),(j,\bar{t}))\in A_m} c_{ij}^k x_{ij}^{kt\bar{t}} + b^k y^k\right] \end{aligned} \tag{4.19}$$

Further, some of the constraints need to be adapted. The transportation capacity constraints are formulated as in (4.20) and replace Constraints (4.4). The platooning constraints take the form of (4.21) instead of (4.5). The design-balance constraints for MVs – Equations (4.6) in the generic model – now need to be applied to each individual MV. To this end, the design-balance constraints are created for each $p \in P$, as shown in Constraints (4.22). This also ensures that each MV returns to the same external zone it departs from. Finally, the domain of the MV service variables is set to be binary in (4.23), in contrast to the integer domain in the generic model.

$$\sum_{k \in K} x_{ij}^{kt\bar{t}} v^k \leq \sum_{p \in P} u_M m_{ijp}^{t\bar{t}} + u_A a_{ij}^{t\bar{t}}, \qquad \forall ((i,t),(j,\bar{t})) \in A \setminus A_h \tag{4.20}$$

$$a_{ij}^{t\bar{t}} \leq n_{ij}^P \sum_{p \in P} m_{ijp}^{t\bar{t}}, \qquad \forall ((i,t),(j,\bar{t})) \in A_m : (i,j) \in A_M \tag{4.21}$$

$$\sum_{((j,\bar{t}),(i,t)) \in A} m_{jip}^{\bar{t}t} = \sum_{((i,t),(j,\bar{t})) \in A} m_{ijp}^{t\bar{t}}, \qquad \forall (i,t) \in N, p \in P \tag{4.22}$$

$$m_{ijp}^{t\bar{t}} \in \{0,1\}, \qquad \forall ((i,t),(j,\bar{t})) \in A, p \in P \tag{4.23}$$

Chapter 5

Study of problem characteristics

In this chapter, we conduct a set of computational experiments to study the functionality of the model and to gain insights into the structure of SNDMAF solutions. Therefore, we conduct an analysis in which we vary the heterogeneous infrastructure, the demand pattern, and the ratio of fixed costs between MVs and AVs. These problem characteristics, which we depict using different instance settings, are expected to have an impact on the strategies to use platooning, the level of consolidation, and the fleet assignment.

The chapter is structured as follows. In Section 5.1, we describe the instances and the experimental setup. Section 5.2 presents a structured analysis of the results of the computational study. Finally, the implications of the study are assessed in Section 5.3.

5.1 Instances

We now define the problem instances for the computational study. Each instance is composed of an infrastructure, a demand pattern, and a fixed cost ratio. In Table 5.1, we give an overview of the various settings we consider for each of these three categories. We consider four infrastructure settings, two demand patterns, and eleven fixed cost ratios. For each possible combination between the three categories' manifestations, we randomly generate five demand allocations which include the origin, the destination, and the volume of commodities. In the following, we describe the categories and their manifestations in more detail.

Table 5.1: Instance settings evaluated in the experiments.

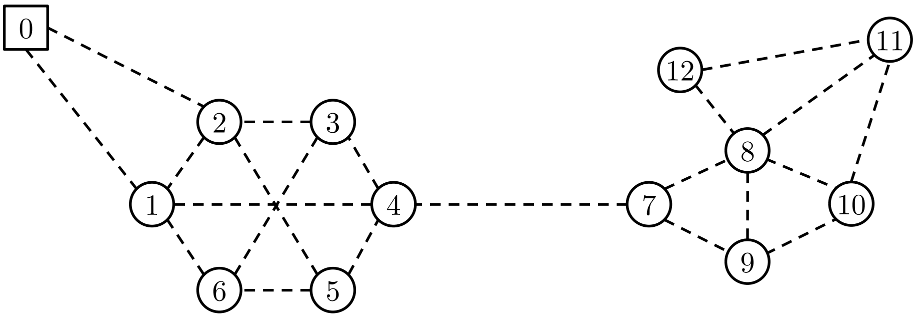
Infrastructure	Demand pattern	Fixed cost ratio
MV only	next-day delivery (NDD)	$FCR = 1.0$
corridor	same-day delivery (SDD)	$FCR = 1.1$
artery		...
nanny		$FCR = 2.0$

5.1.1 Infrastructure

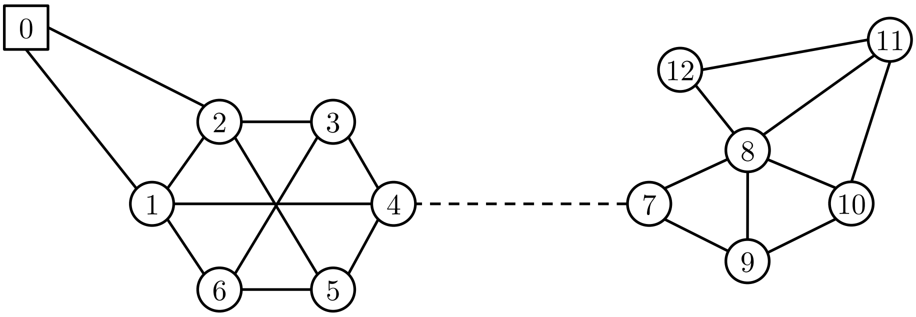
First, we describe how we generate the different types of heterogeneous infrastructure that are depicted in Figure 5.1. Each setting is based on one common underlying physical network which includes the locations of one external zone and twelve satellites. Later in Section 5.2.7, we describe a modified version of this network to analyze the effect of multiple external zones. Physical arcs connect the locations in both directions with travel times that are symmetric in both directions. The twelve satellites are grouped into two clusters that both comprise six satellites. These clusters are only connected with each other via an arc between satellites 4 and 7. The single external zone is connected with the left cluster only via two arcs, such that vehicles need to cross this cluster to reach the right cluster.

From the underlying physical network, we derive the heterogeneous infrastructure by determining a set of physical arcs as AV arcs. For the instances with AVs, we define physical networks that each resemble one of the three heterogeneous infrastructure types previously distinguished in Section 3.3. Figure 5.1 illustrates the layout of the physical networks based on the specific heterogeneous infrastructure. The locations are numbered with external zones depicted as squares and satellites depicted as circles. MV arcs are drawn as dashed lines and AV arcs are drawn as solid lines.

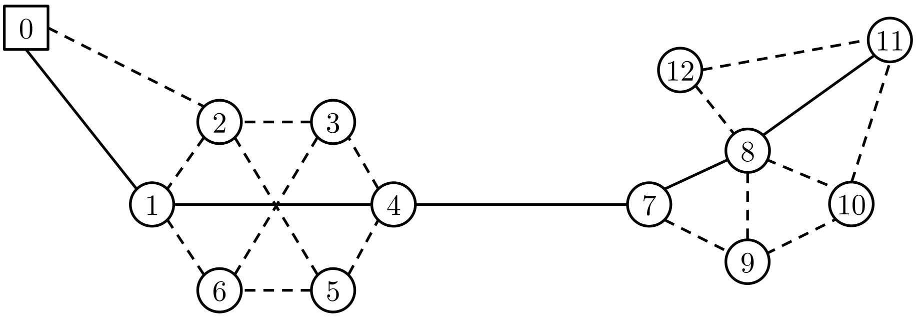
The four networks are described in the following.



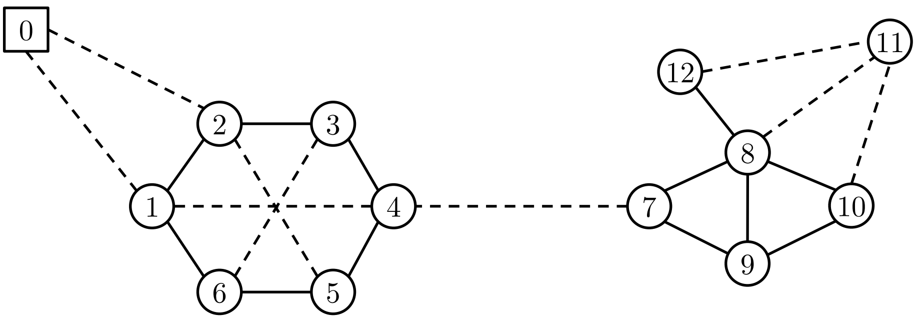
(a) MV only infrastructure



(b) corridor infrastructure



(c) artery infrastructure



(d) nanny infrastructure

Figure 5.1: Physical networks based on heterogeneous infrastructure, adapted from Scherr et al. (2019).

- Figure 5.1a illustrates the physical network for the *MV only* setting that does not include any AV arcs. This infrastructure can basically be denoted as homogeneous compared to the heterogeneous types with MV and AV arcs. We further prohibit the use of AVs by using the fleet management constraint (4.18) and setting the number of AVs to $n_A = 0$. This setting resembles the conventional SND problem with a homogeneous fleet of MVs only, therefore representing a benchmark for the impact of AVs in a mixed autonomous fleet.

- Figure 5.1b displays the physical network for the *corridor* infrastructure. In this heterogeneous infrastructure, the arc connecting the two clusters is the only MV arc. Thus, it is expected that MVs predominantly move in proximity to the corridor and transfer AVs between the two clusters.

- The physical network we use to represent the *artery* infrastructure is depicted in Figure 5.1c. AVs are only allowed to travel on the artery that crosses the network. The artery leads from the external zone on the shortest path connecting the locations 0–1–4–7–8–11 towards the furthermost satellite 11 and backwards. In this way, the artery also crosses the corridor between the two clusters. As AVs can travel through the network on AV arcs, we expect that platooning is not encouraged by this layout.

- Figure 5.1d shows the physical network for the *nanny* infrastructure. In the left cluster of satellites, AV arcs connect the complete outer ring. Platooning can be used as a shortcut in this cluster. In the right cluster, the arcs located near the corridor are AV arcs. Platooning by an MV is required for bridging the corridor between the two clusters or for reaching the outermost satellite 11. Also, there is no AV arc between the external zone and the satellites in this physical network. We expect that platooning is used in parts of this fragmented network.

The physical network is transformed into a time-expanded network with a discretization of $\Delta = 10$ minutes. We examine a planning horizon with a length of 5 hours, i.e., $t_{max} = 300$ minutes. Thus, a total of 30 time periods t are considered with $t \in T =$

$\{0, 10, \ldots, t_{max}\}$. Locations are replicated according to the interval length over the planning horizon to yield the nodes of the time-expanded network. The physical arcs are transformed into arcs of the time-expanded network. Their length is obtained by rounding up the ratio between the travel time τ_{ij} and the length of the time intervals $\Delta = 10$ minutes.

5.1.2 Demand pattern

Each instance requires a commodity demand that needs to be transported. We first generate five demand allocations, in which we consider 24 commodities k that are each assigned a volume $v^k \in \{1, 2, \ldots, 5\}$ in discrete volume units. The origin o^k of each commodity is defined as the external zone together with a departure time at which the commodity is available. The storage capacity of the external zone, denoted by parameter u_E, is not limited within the experiments such that there are no restrictions for holding commodities at the external zone. Each commodity's destination d^k is defined as a satellite and an arrival time. The destinations are randomly generated such that each of the twelve satellites receives two commodities within the planning horizon.

The two demand patterns which we evaluate both use the same commodity destinations $d^k = (i_d^k, t_d^k)$ and volumes v^k, however, the departure times t_o^k of the commodities at their origin external zone i_o^k are different.

- The first demand pattern resembles a *next-day delivery (NDD)* setting. In the NDD demand pattern, the complete supply of commodities is available at the external zone at the beginning of the planning horizon, such that the departure time of each commodity at its origin is $t_o^k = 0$. This demand pattern mirrors a traditional approach, in which inter-city transportation towards the external zone is performed at the day before or over night, such that all commodities are available at the start of the day. We assume that the NDD pattern enables a high level of consolidation at the external zone.

- The second demand pattern resembles a *same-day delivery (SDD)* setting. Here, the commodities arrive at the external zone at different departure times within the planning horizon. We generate the SDD instances in a way that the commodities arrive exactly 90 minutes ahead of their arrival time t_d^k, i.e., the departure time at the origin is $t_o^k = t_d^k - 90$. This demand pattern mirrors an approach in which commodities arrive at the external zone at multiple times over the course of the day, e.g., because shorter delivery times are required or the supply towards the external zone is less consolidated. We surmise that the SDD demand pattern requires more frequent returns to the external zone.

Applying the NDD and SDD demand pattern to each of the five demand allocations results in a total of ten commodity demand instances which we consider in the experiments.

5.1.3 Fixed cost ratio

In the experiments, we assume that the fixed costs of MVs, f_M, and those of AVs, f_A, are different. As the fixed cost of an MV also includes the wages for employing a driver, we assume that the fixed cost of an MV is at least as high as the fixed cost of an AV, which generally does not require a driver. To quantify the relation between the fixed cost of MVs and AVs, we introduce the *fixed cost ratio* $FCR = f_M/f_A$. In the experiments, we increase the fixed cost of MVs, f_M, to test the different fixed cost ratios $FCR = \{1.0, 1.1, \ldots, 2.0\}$. We do not enforce a limit on the number of MVs and AVs via the fleet management constraints in the experiments such that the fixed cost ratio directly impacts the size and mix of the mixed autonomous fleet.

Besides the fixed costs and the ability to travel on specific arcs of the network, the two vehicle types are identical in their characteristics. The service costs and transportation costs are the same for MVs and AVs. Outsourcing is not considered in these experiments. The parameters determining the transportation capacity of MVs and AVs are set to $u_M = u_A = 10$ volume units. We set the platoon capacity to $n_{ij}^P = 2$ on all physical arcs $(i, j) \in A_{ph}$, thus we will refer to this parameter simply as n^P from here on. Reports indicate that such a platoon length of three vehicles in total can be considered as reasonable in urban traffic, as the maximum vehicle length that is considered at inner-city traffic lights in Germany is 15 meters (Forschungsgesellschaft für Straßen- und Verkehrswesen, 2010).

In conclusion, we generate a total number of 440 instances for this study. Each instance results in an integer program of a challenging size with over 4,000,000 variables and over 4,000,000 constraints. Note that the exact number of constraints depends on the number of MV arcs in the heterogeneous infrastructure, as the platooning constraints (4.5) are only applied to MV arcs.

5.2 Results

We conduct the experiments by solving the respective instances with the commercial solver CPLEX in version 12.8 (IBM, 2020) on a 32-core machine with 2.00 GHz Intel Xeon X7550 CPUs. The respective models were created using the CPLEX Concert Technology library for C++. We set the optimality gap tolerance to 1% and impose a CPU time limit of 400,000 seconds, i.e., around 111 hours. The solver run terminates if either one of these two criteria is met. In the following, we report the results from the computational study.

5.2.1 Computational performance

First, we analyze the computational performance of CPLEX in solving the SNDMAF instances to optimality. We particularly aim to assess the impact on the runtime of considering a mixed fleet of MVs and AVs compared to solving MV only instances. The MV only instances represent a conventional SND model without the requirements to decide between MVs and AVs to perform services and to synchronize them for platooning. In Figure 5.2, we depict the runtime in CPU hours of the "MV only" instances and the instances using the mixed autonomous fleet, denoted as "MV+AV". The runtime is averaged over all five demand allocations, eleven fixed cost ratios, and, in the case of the MV+AV instances, all three infrastructure settings. In the figure, we also distinguish between the NDD and the SDD demand pattern.

The results show that the MV+AV instances are computationally challenging and require, in total, more than eight times the runtime than the MV only instances. The

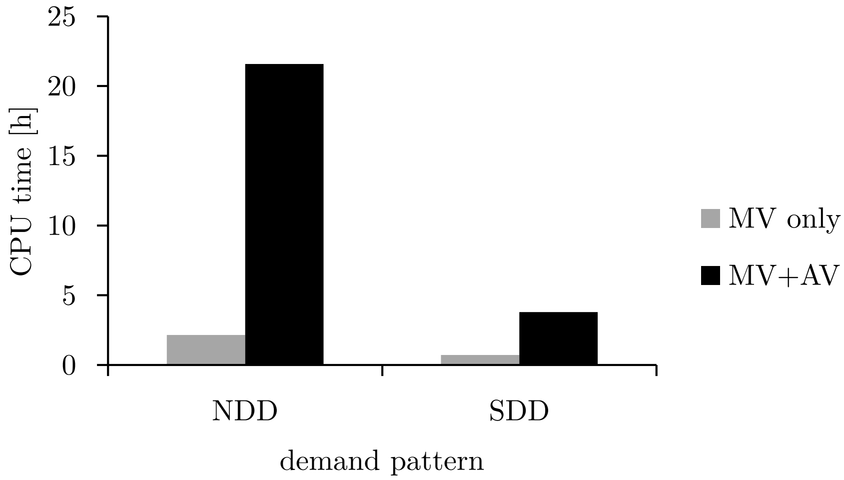


Figure 5.2: Runtime of MV only instances and instances with MVs and AVs.

instances with the NDD demand pattern reveal an even larger difference with a factor of around ten compared to the SDD instances with a factor of around five between MV+AV and MV only instances. This may be caused by the larger number of synchronization and consolidation opportunities in the NDD demand pattern, in which the delivery time windows are larger because of the earlier availability of the commodities. CPLEX was able to solve all MV only instances to optimality within the tolerance, whereas 3% of the MV+AV instances reach the CPU time limit with a maximum gap of 5% reported over all instances. As we always report the best primal solution obtained after the runtime limit in the following, we thus underestimate the potential of considering a mixed autonomous fleet compared to an MV only setting.

5.2.2 Cost sensitivity

In this section, we analyze the impact of deploying AVs on the total costs of an LSP. Therefore, we compare the total costs obtained as the objective function value between the MV+AV settings, given different heterogeneous infrastructure, and the MV only setting. In Figures 5.3 and 5.4, we show the average cost savings that can be obtained from considering AVs in the corridor, artery, and nanny infrastructure. We depict the different fixed cost ratios on the x-axis by setting $FCR = \{1.0, 1.1, \ldots, 2.0\}$. As there are no AVs considered in the MV only setting, increasing the fixed cost ratio simply results in an MV fixed cost of the same value as in the MV+AV settings. We distinguish between results for the NDD demand pattern presented in Figure 5.3 and results for the SDD demand pattern presented in Figure 5.4.

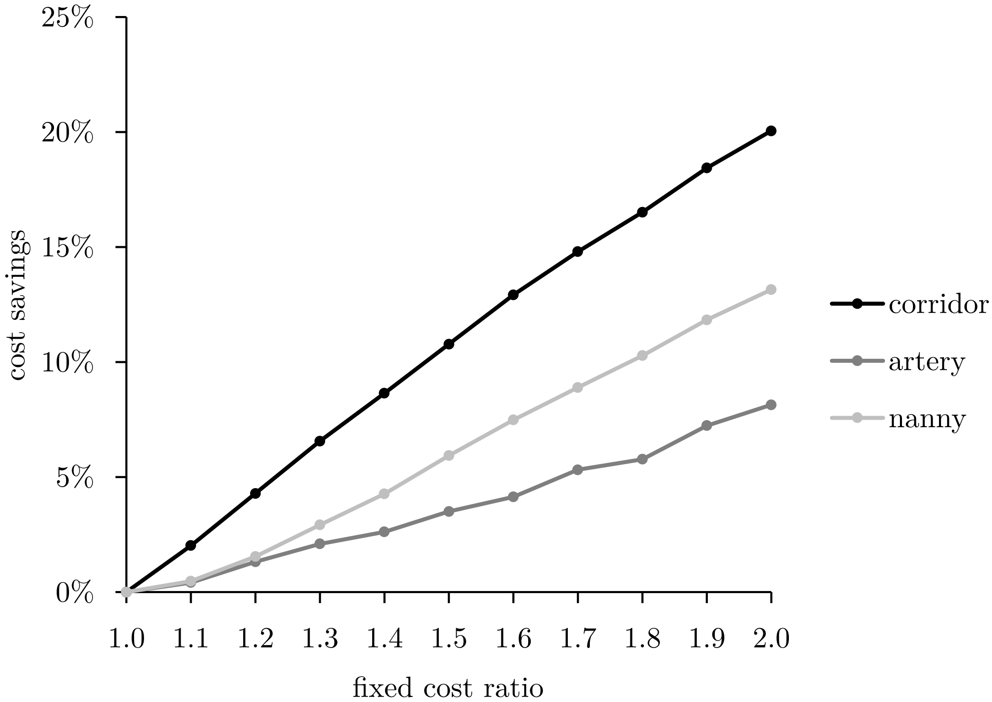


Figure 5.3: Cost savings with AVs in different infrastructures compared to MV only setting with NDD demand pattern, adapted from Scherr et al. (2019).

The first observation is that considering AVs leads to average cost savings across all fixed cost ratios and infrastructures. Also, cost savings for each infrastructure increase strictly monotonically with growing fixed cost ratios. As expected, the higher the fixed cost is for MVs, the more savings can be obtained from substituting MVs with AVs. We analyze the underlying reason for the cost savings in the next section in which we investigate the fleet size and mix. The increase is steeper for the NDD instances, leading to cost savings of up to 20% with $FCR = 2.0$, compared to a maximum of 16% cost savings for the SDD instances. The reason for this difference is possibly due to more extensive opportunities for consolidation in the NDD demand pattern.

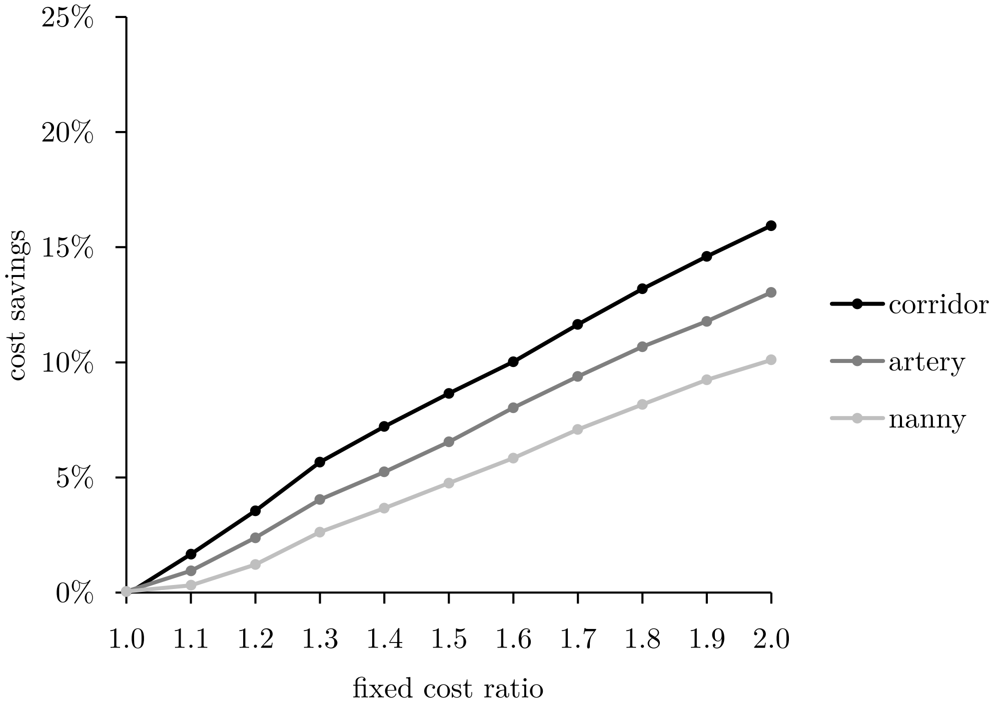


Figure 5.4: Cost savings with AVs in different infrastructures compared to MV only setting with SDD demand pattern, adapted from Scherr et al. (2019).

We can also observe differences between the three types of heterogeneous infrastructure regarding the cost savings. The largest cost savings in both demand patterns are obtained with the corridor infrastructure. Since this infrastructure includes the largest share of AV arcs across the network, it encourages fleets with more AVs. The artery infrastructure yields the smallest savings in the NDD demand pattern, whereas it yields the second largest savings in the SDD demand pattern, with values close to those of the corridor infrastructure. In the SDD demand pattern with small time windows for the delivery of commodities, AVs may benefit from using the artery to return to the external zone quickly and without the need for platooning. Thus, the artery infrastructure may show a greater sensitivity to the demand pattern. The nanny infrastructure shows a contrary effect, as the cost savings appear to be less sensitive to the given demand pattern. For both the NDD and SDD demand pattern, the obtained cost savings in the artery and nanny infrastructure are closely related to those of the corridor infrastructure, albeit smaller.

5.2.3 Fleet size and mix

To explain the reasons for the cost savings obtained in the previous section, we now analyze the fleet size and, particularly, the fleet mix between MVs and AVs. In Figures 5.5 and 5.6, we report the average number of MVs and AVs that is assigned for each fixed cost ratio and infrastructure. Again, we distinguish between the NDD demand pattern depicted in Figure 5.5 and the SDD demand pattern depicted in Figure 5.6. The size of each bar in the charts represents the fleet size as the total number of MVs and AVs, with the respective shade indicating the infrastructure. Further, a bar consists of a filled part at the bottom, depicting the number of MVs, and a framed part at the top, depicting the number of AVs. Note that there may exist symmetric solutions with the same objective function value but with a different fleet assignment for some instances, particularly those with $FCR = 1.0$. We consider the fleet size and mix of the best found solution by the solver and report the average values over five demand allocations for each bar in the chart.

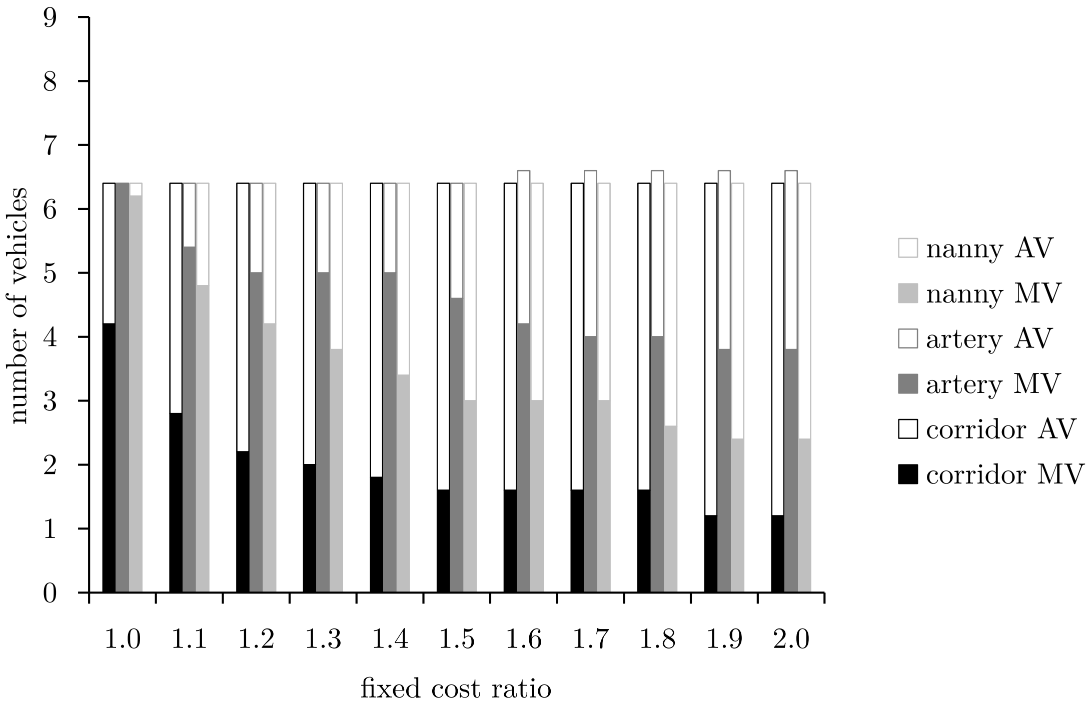


Figure 5.5: Fleet size and mix with NDD demand pattern for different infrastructures and fixed cost ratios, adapted from Scherr et al. (2019).

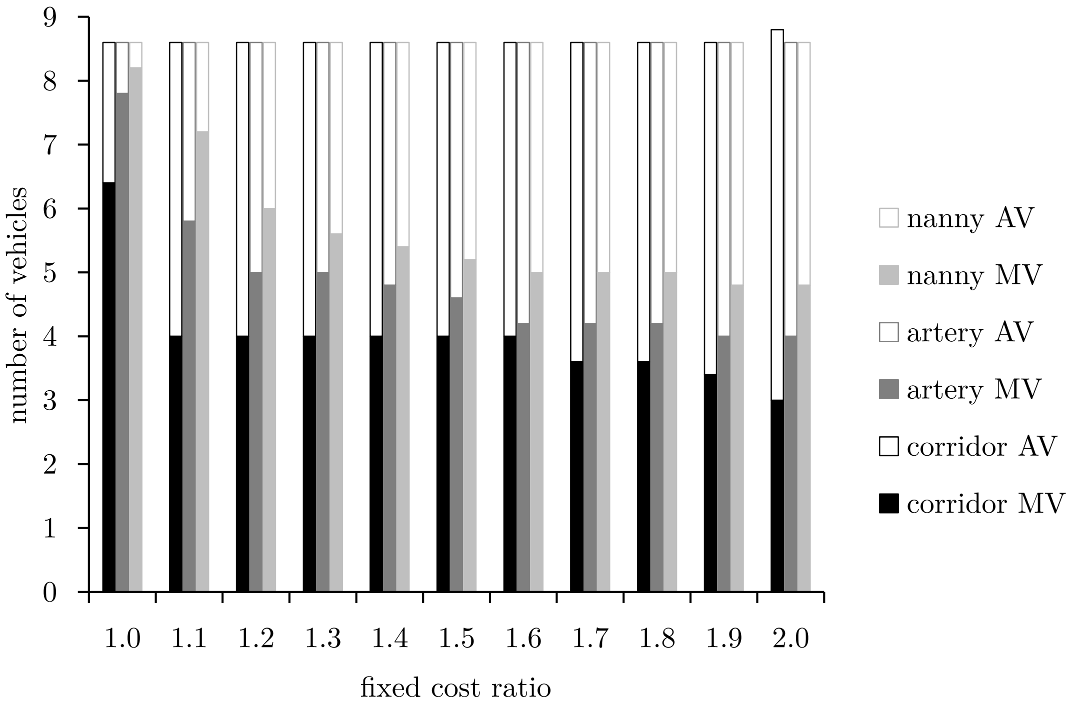


Figure 5.6: Fleet size and mix with SDD demand pattern for different infrastructures and fixed cost ratios, adapted from Scherr et al. (2019).

The first general observation from both figures is that the fleet size is approximately constant for different types of infrastructure and fixed cost ratios in each of the two demand patterns, respectively. Although not depicted in the figures, the total number of MVs and AVs is similar to the number of MVs in the MV only instances. This means that – regardless of the fleet mix – the reported fleet size is necessary to provide sufficient transportation capacity for the commodity demand. The second general observation is that the fleet mix changes when increasing the fixed cost ratio. Over all infrastructure settings and both demand patterns, the number of MVs monotonically decreases and the number of AVs monotonically increases with a growing fixed cost ratio. The substitution of MVs with AVs provides an explanation for the cost savings assessed previously.

The results for the NDD demand pattern in Figure 5.5 show that the corridor infrastructure requires the smallest number of MVs. Only 1.2 MVs on average are assigned to the fleet with $FCR = 2.0$. The nanny and the artery infrastructure require a larger number of MVs than the corridor infrastructure. Nevertheless, the fleet mix changes with increasing the fixed cost ratio with a similar slope for all three types of infrastructure.

The results for the SDD demand pattern in Figure 5.6 confirm this behavior regarding the fleet mix. Compared to the NDD instances, fulfilling the demand of the SDD instances on average requires around a third of additional vehicles, i.e., a larger fleet size. The additional assigned vehicles are mainly MVs in the corridor and nanny infrastructure, as they are more flexible in terms of traveling through the network. In the artery infrastructure, however, more AVs are assigned in the SDD demand pattern which explains the larger cost savings presented in the previous section.

5.2.4 Infrastructure usage of MVs and AVs

In this section, we analyze the use of MV and AV services in the SNDMAF solutions obtained from the experiments. We particularly focus on the services that denote traveling between locations, as these services have an impact on the usage of public infrastructure and, thus, traffic in the city. For the sake of simplicity, we entitle these services that are installed on movement arcs $((i,t),(j,\bar{t})) \in A_m$ as *moves* in the remainder of this chapter. To measure the impact on urban traffic in more detail, we specifically analyze the travel time that vehicles consume in moves rather than the sheer number of services. Idling of vehicles, which is denoted by services installed on holding, waiting, or auxiliary arcs, takes place in the LSP's dedicated parking areas at external zones and satellites. Thus, we conclude that idling does not have a relevant impact on traffic.

The number of *MV* and *AV moves* can be directly extracted from the solution values of variables $m_{ij}^{t\bar{t}}$ and $a_{ij}^{t\bar{t}}$, respectively, on a movement arc. Since platoons are not explicitly modeled as variables, we derive the use of platooning from MV and AV services that synchronize on the same arc. Following this, we denote *MV moves in platoons* on an arc as the value $m_{ij}^{t\bar{t}P}$ based on Equation (5.1). The equation defines the minimum number of MV services that are required to platoon a number of AVs given the platoon capacity n^P. Only this number of MVs is assumed to lead platoons on the arc. *AV moves in platoons* on an arc are denoted as the value $a_{ij}^{t\bar{t}P}$ and can be derived with Equation (5.2). This equation determines the number of AVs that are able to travel in platoons on an arc given the number of MV services and their associated platoon capacity installed on that arc. Thus, AV moves are only counted as platoon moves if the leading MVs provide sufficient platoon capacity.

$$m_{ij}^{t\bar{t}P} = \min\left(m_{ij}^{t\bar{t}}, \left\lceil \frac{a_{ij}^{t\bar{t}}}{n^P} \right\rceil\right), \qquad \forall((i,t),(j,\bar{t})) \in A_m \tag{5.1}$$

$$a_{ij}^{t\bar{t}P} = \min\left(a_{ij}^{t\bar{t}}, m_{ij}^{t\bar{t}} n^P\right), \qquad \forall((i,t),(j,\bar{t})) \in A_m \tag{5.2}$$

In Figures 5.7 and 5.8, we analyze the total travel time in minutes consumed by MVs and AVs within the planning horizon. We report results for each infrastructure, including the MV only setting, and for three distinct fixed cost ratios to observe the effect of various fleet mixes. Since the fixed cost ratio does not show an effect on the total travel time in the MV only setting, we only report results for $FCR = 2.0$ here. For the corridor, artery, and nanny infrastructure, the left bar represents the results for $FCR = 1.0$, the middle bar for $FCR = 1.5$, and the right bar for $FCR = 2.0$. In each bar, we distinguish for each vehicle type between moves outside of platoons, denoted as "MV" and "AV", and moves in platoons, denoted as "MV in platoon" and "AV in platoon". Again, Figure 5.7 presents average results for all demand allocations with the NDD demand pattern and Figure 5.8 presents results for those with the SDD demand pattern.

A general observation from both figures is that substituting MVs with AVs with increasing fixed cost ratios – as we noted in the previous analysis of the fleet mix – only results in a minor increase of the fleet's total travel time. If each platoon would be considered as one coherent vehicle – which can be discussed given the smaller headway between vehicles – AV moves in platoons could be ignored in each bar of the chart. Based on this assumption, the total travel time and, thus, the impact on traffic, would be smaller in solutions with extensive use of platooning. Also, we observe that not only the number of AVs increases with a growing MV fixed cost but also the travel time consumed by AV moves, both in and outside of platoons. Analogously to the findings regarding the

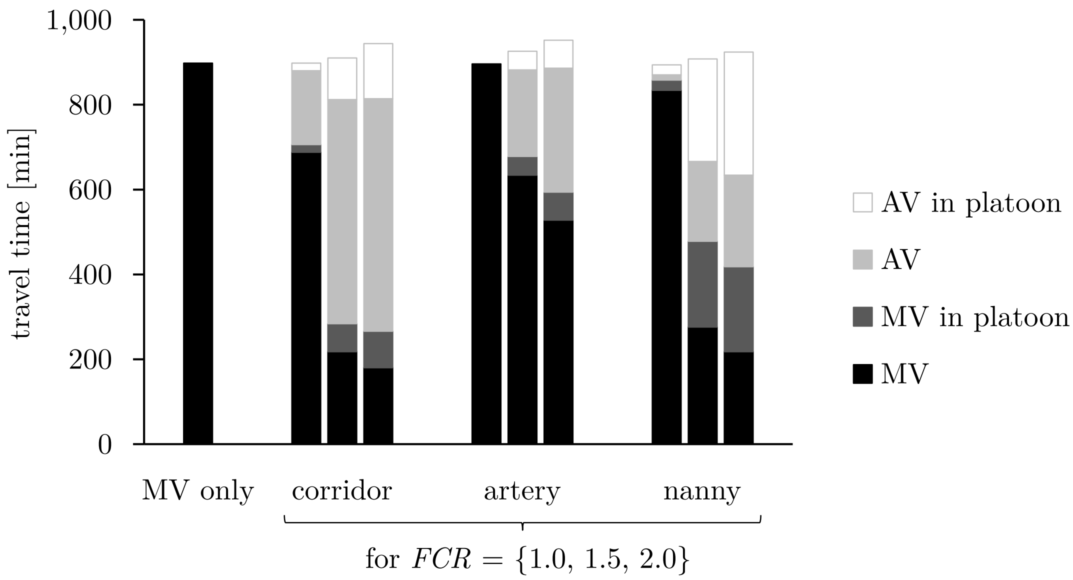


Figure 5.7: Travel time by vehicle type with NDD demand pattern for different infrastructures and fixed cost ratios, adapted from Scherr et al. (2019).

fleet size, the SDD demand pattern requires around an additional third of the travel time compared with the NDD demand pattern across all infrastructures and fixed cost ratios.

Several observations can also be derived from comparing the three types of heterogeneous infrastructure in the results depicted in both figures. Although the total travel time is similar in all settings, the distribution between MVs and AVs as well as the use of platooning vary widely. This observation is clearest with $FCR = 2.0$, as the use of AVs is encouraged the most in this setting. In the corridor infrastructure, the shortest travel time is allocated to MV moves because AVs can travel in large parts of the network independently. In the artery infrastructure, the AVs' proportion of the travel time depends on the demand pattern. With the SDD demand, around half of the travel time is contributed by AVs, which is similar to the distribution in the corridor infrastructure. With the NDD demand, however, AVs only contribute around a third of the total travel time. While the share of moves in platoons is relatively small in the corridor and artery infrastructure, the nanny infrastructure encourages the use of platooning. In the NDD demand pattern, MVs lead platoons around half of their travel time and AVs even spend more travel time as platoon followers than outside of platoons.

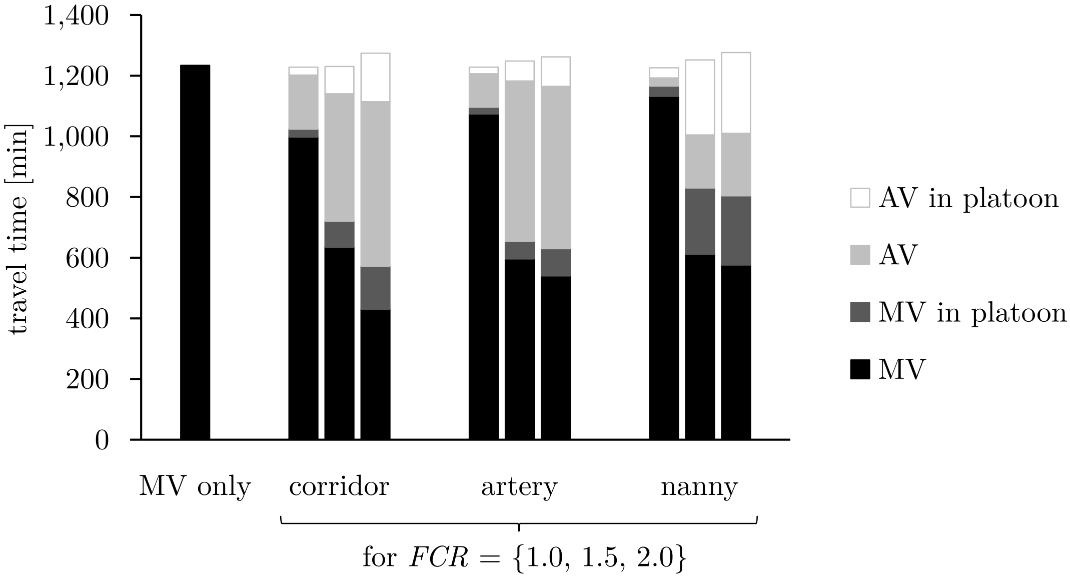


Figure 5.8: Travel time by vehicle type with SDD demand pattern for different infrastructures and fixed cost ratios, adapted from Scherr et al. (2019).

5.2.5 Platoon length

In this section, we investigate the platoon length, i.e., the number of vehicles in a platoon, in more detail. In an SNDMAF solution, each platoon consists of exactly one MV and a number of AVs, which is restricted by the platoon capacity n^P. For this reason, we measure the average number of AVs per platoon which we obtain from dividing the total number of AV moves in platoons by the total number of MV moves in platoons, i.e, with $\sum_{((i,t),(j,\bar{t}))\in A_m} \frac{a_{ij}^{t\bar{t}P}}{m_{ij}^{t\bar{t}P}}$.

In Figures 5.9 and 5.10, we report the average number of AVs per platoon with $FCR = 2.0$, as this fixed cost ratio has shown to promote an extensive use of platooning. Both figures show results for the three types of heterogeneous infrastructure. Figure 5.9 includes the results for the NDD demand pattern and Figure 5.10 those for the SDD demand pattern. The platoon capacity has been set to $n^P = 2$ in all previous experiments, which allows a maximum of two AVs per platoon. To investigate whether this restriction compromises the desired platoon length by the LSP, we conduct experiments with the parameter set to $n^P = 3$ and $n^P = 4$ to permit longer platoons.

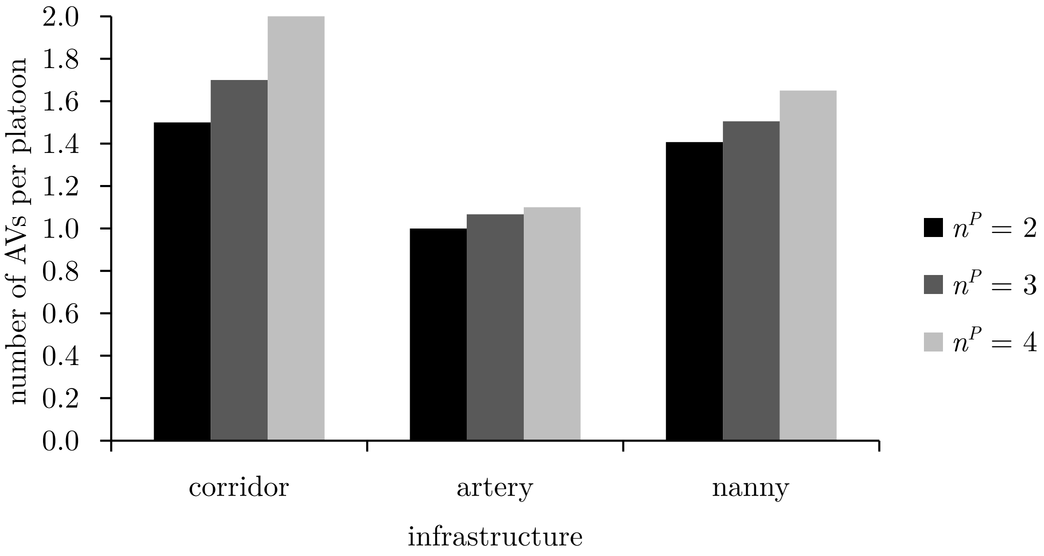


Figure 5.9: Platoon length in NDD demand pattern for different platoon capacities and infrastructures with $FCR = 2.0$, adapted from Scherr et al. (2019).

The results for the NDD demand pattern, depicted in Figure 5.9, show that longer platoons are formed in the corridor and nanny infrastructure than in the artery infrastructure. Also, increasing the platoon capacity can be utilized by the LSP in those two infrastructure types in which platooning proves to be useful. The corridor infrastructure, in which platooning is used only on the corridor arc to traverse AVs between the two clusters, shows the largest sensitivity to relaxing the platoon capacity restriction. However, the average platoon does not contain more than two AVs even after setting $n^P = 4$. In the artery infrastructure, platoons are typically composed of one MV and one AV. Thus, increasing the platoon capacity only shows a minor impact.

The average platoon length in the SDD demand pattern, depicted in Figure 5.10, is considerably smaller than in the NDD demand pattern. Platoons on average consist of only around one AV. Also, the sensitivity to increasing the platoon capacity can be considered as negligible in these instances. The LSP can utilize longer platoons only in the corridor infrastructure to a small extent. The tight time windows in the SDD demand pattern are assumed to reduce waiting times of vehicles, thus reducing the potential for vehicles to

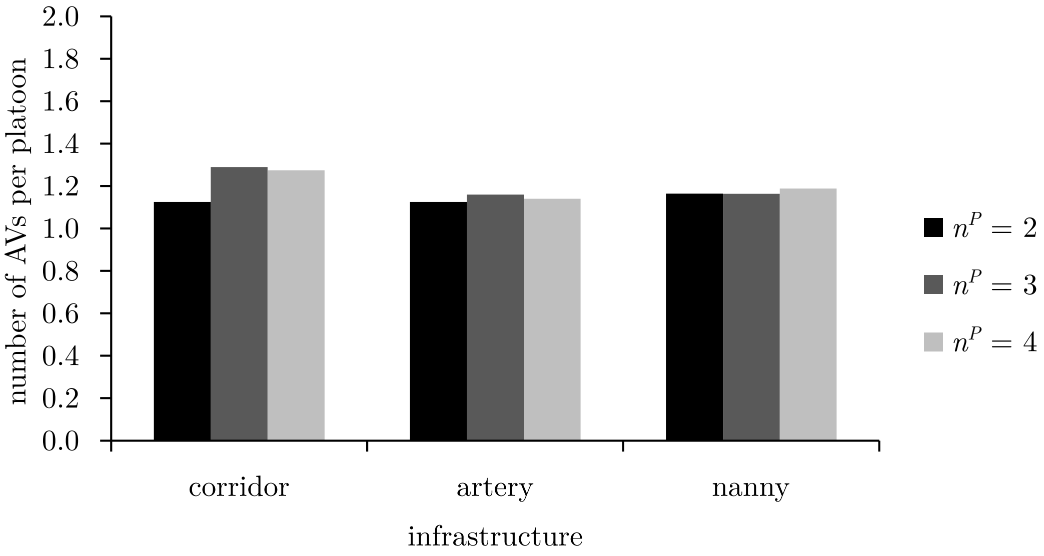


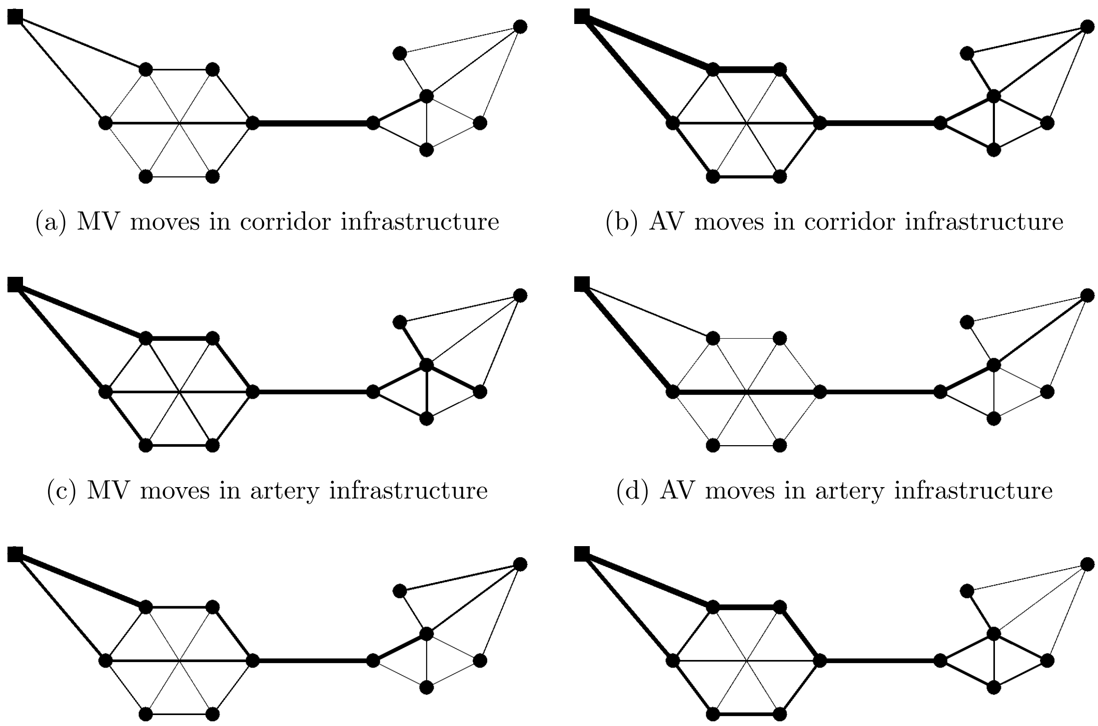
Figure 5.10: Platoon length in SDD demand pattern for different platoon capacities and infrastructures with $FCR = 2.0$, adapted from Scherr et al. (2019).

consolidate and synchronize to form platoons.

5.2.6 Insights into solutions

In this section, we investigate the solutions in more detail with a particular focus on the roles of MVs and AVs given different heterogeneous infrastructure. We consider the results for the NDD demand pattern and the highest fixed cost ratio $FCR = 2.0$ in this analysis, as these have shown the most pronounced usage of AVs and platooning. To compare the roles of the two vehicle types, we measure the frequency of MV and AV moves in the physical network. The frequency is obtained by considering the sum of all moves of a vehicle type on a physical arc $(i, j) \in A_{ph}$ over all time periods, i.e., $\sum_{t \in T} m_{ij}^{t\bar{t}}$ and $\sum_{t \in T} a_{ij}^{t\bar{t}}$, respectively.

In Figure 5.11, we illustrate the frequency of MV and AV moves for the three types of infrastructure with each one representing a row of the grid. MV moves are visualized in the left column of the grid and AV moves in the right column. In each figure, a physical network is depicted in which the width of each arc corresponds to the frequency of moves on that arc. The wider an arc, the more moves are performed on this arc by the respective vehicle type over the planning horizon. For the sake of simplicity, the frequencies of both directions of one arc, i.e, (i, j) and (j, i), are consolidated such that the aggregated frequency is depicted in these figures.



(a) MV moves in corridor infrastructure (b) AV moves in corridor infrastructure

(c) MV moves in artery infrastructure (d) AV moves in artery infrastructure

(e) MV moves in nanny infrastructure (f) AV moves in nanny infrastructure

Figure 5.11: Frequency of moves in the physical network in NDD demand pattern with $FCR = 2.0$.

In Figure 5.11a, which shows the MV moves in the corridor infrastructure, the role of the MVs is clearly dedicated to platooning AVs on the corridor arc back and forth between the two clusters. The majority of the remaining moves serve the purpose of traveling towards the corridor arc and returning to the external zone from there. Figure 5.11b, which depicts the respective AV moves in the corridor infrastructure, confirms that MVs platoon AVs towards the right cluster of satellites. Otherwise, AVs travel through the network independently from MVs.

Conversely, it can be observed that MV moves are spread over the complete network in the artery infrastructure, as depicted in Figure 5.11c. The reason for this can be obtained from Figure 5.11d, which shows the AV moves in the artery infrastructure. AVs predominantly move on the AV arcs that are part of the artery through the network. AVs supply commodities from the external zone to satellites on the artery, from which MVs perform the deliveries to remote satellites after the commodities have been transshipped.

For the nanny infrastructure, the MV moves are depicted in Figure 5.11e and the AV moves in Figure 5.11f. The distribution of MV and AV moves in the network is not as distinct as in the other two types of infrastructure. Nevertheless, it can be observed that AVs tend to use the outer ring of AV arcs in the left cluster of satellites. Platooning is extensively used on both MV arcs that connect the external zone with the left cluster and on the corridor arc connecting the two clusters.

5.2.7 Fixed external zone assignment of MVs

At the end of this results section, we investigate the impact of multiple external zones and how MVs are assigned to them. To this end, we conduct additional experiments, as the physical network used for all previous experiments only contains one external zone, see Figure 5.1. We modify the physical network with the corridor infrastructure to include an additional external zone that is connected with the right cluster of satellites. Figure 5.12 displays the physical network with two external zones, which we entitle *EZ 2*. Correspondingly, the physical network with one external zone is called *EZ 1* in this section.

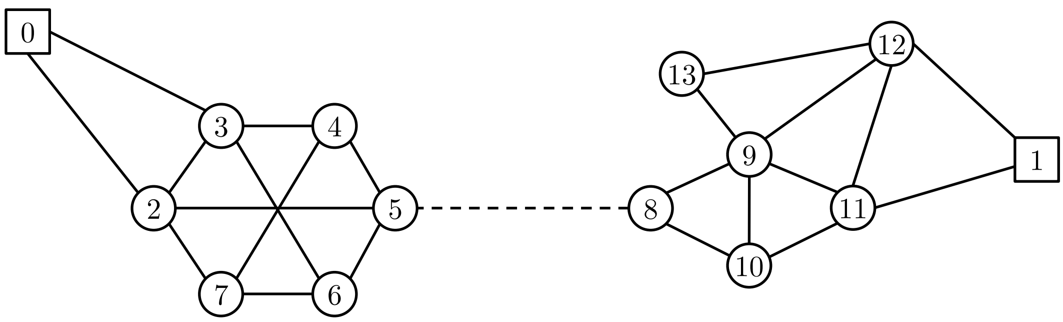


Figure 5.12: Physical network with two external zones and corridor infrastructure, adapted from Scherr et al. (2019).

We conduct a small computational study on the EZ 2 network with the same parameter settings as in the previous experiments. We set the fixed cost ratio to a medium setting with $FCR = 1.5$. The demand allocations are modified such that one of the two commodity deliveries each satellite receives originates from the additional external zone. Otherwise, the commodity origins and volumes are unaltered compared to the EZ 1 experiments. On the EZ 2 network, we also analyze the impact of a fixed assignment of MVs to a specific external zone. For this reason, we consider the model modification described in Section 4.3.3, in which each MV is considered individually and needs to return to the same external zone it departs from at the end of the planning horizon. This setting is denoted as *Fixed EZ 2*.

In Table 5.2, we report results for the EZ 1, EZ 2, and Fixed EZ 2 settings for each demand instance (Column "Demand"). A demand instance is composed of a demand pattern (NDD or SDD) and a numbered demand allocation. We report the CPU runtime in seconds until an optimality gap of 1% is reached for all three settings in Columns "1% gap". However, not all instances could be solved by CPLEX within the time limit of 400,000 seconds in the EZ 2 and Fixed EZ 2 settings. Thus, we additionally report the time at which the final best primal solution (Columns "Best") and the time at which an optimality gap smaller than 5% (Columns "5% gap") is obtained. If the runtime limit is reached for an instance, we report the remaining gap in % instead.

We observe that solving the EZ 2 instances generally requires more computational effort than EZ 1. When comparing the EZ 2 with the Fixed EZ 2 setting, it can be observed that the fixed assignment of MVs further increases the runtime. This is not surprising as modeling each MV individually rather than considering aggregated flows increases the model size. Although CPLEX does not find the optimal solution for every instance within the time limit, it is still able to find solutions of reasonable quality for all of the instances.

Table 5.2: Runtime in CPU seconds or gap in % at runtime limit for physical network with two external zones and fixed external zone assignment of MVs, adapted from Scherr et al. (2019).

Demand	EZ 1	EZ 2			Fixed EZ 2		
	1% gap	1% gap	Best	5% gap	1% gap	Best	5% gap
NDD_0	9,806	39,906	36,751	36,751	1.28%	241,658	185,865
NDD_1	61,406	4.21%	392,048	392,048	5.23%	301,291	5.23%
NDD_2	15,982	37,260	37,260	25,755	121,326	82,013	52,283
NDD_3	101,020	2.46%	59,284	59,284	3.93%	70,542	235,461
NDD_4	20,087	367,524	254,721	56,911	2.68%	374,602	82,265
SDD_0	675	1,707	1,707	1,707	10,279	10,279	7,556
SDD_1	15,596	125,168	2,969	5,431	2.82%	59,096	33,684
SDD_2	2,761	2,668	2,668	1,400	8,165	8,165	7,105
SDD_3	1,302	29,105	4,073	5,535	1.51%	6,797	19,517
SDD_4	1,803	5,092	1,091	5,092	7,767	4,684	2,228

5.3 Implications

The computational study illustrates the correctness and versatility of the proposed model for the service network design problem with mixed autonomous fleets. Medium-sized instances can be solved with an off-the-shelf solver applied to the mathematical model as formulated in Chapter 4. Results show that the obtained solutions adapt to a variety of instance settings with regard to the heterogeneous infrastructure, the demand pattern, and the fixed cost ratio of MVs and AVs. Modifying these instance settings shows to have an impact on the potential cost savings, the fleet mix, and on the strategies of using MVs and AVs to perform transportation services as well as coordinating them in platoons.

Based on the results of this computational study, it can be concluded that the potential cost savings obtained from using AVs largely depend on the fixed cost ratio between MVs and AVs. An assumption of this study is that deploying an AV is generally at most as expensive as deploying an MV, which requires additional costs for hiring a human driver to maneuver the MV. Since AVs are only able to travel on parts of the network independently from MVs, the heterogeneous infrastructure limits the number of MVs that can be substituted with more cost-efficient AVs. A general observation from the results is that the more AV arcs are considered in a heterogeneous infrastructure, the larger cost savings can be obtained.

Associated with the cost savings is the impact on the fleet mix that is determined by the number of MVs and AVs. While the heterogeneous infrastructure and the demand pattern restrict the number of assigned AVs in the SNDMAF solutions, one MV is typically substituted by one AV over all instances settings. Thus, the fleet size, composed of the total number of MVs and AVs, is generally constant over all evaluated fixed cost ratios and types of heterogeneous infrastructure. Only the demand pattern has an effect on the fleet size, with the SDD demand pattern requiring a larger fleet. The additionally required vehicles compared to the fleet for the NDD demand pattern are mainly MVs.

The strategies for using the different vehicle types widely vary depending on the heterogeneous infrastructure, the demand pattern, and the fleet mix, which is in turn impacted by the fixed cost ratio. While the layout of AV arcs determines the parts of the network in which AVs predominantly perform transportation services, each heterogeneous infrastructure exposes different strategies for using platooning outside of AV arcs. In the corridor infrastructure, platooning is used only locally to transfer AVs between satellite clusters. The platoon length is typically large such that the platoon capacity of each of the few MVs is nearly exhausted. In the artery infrastructure, platooning is mostly used to extend the transportation capacity of MVs, as AVs can travel through the network independently. In the nanny infrastructure, platooning is used extensively on multiple parts of the network, e.g., to reduce the travel times of AVs by taking shortcuts in platoons.

Although this computational study provides insights into the general behavior of the model concerning different instance settings and the structure of solutions, there are some limitations regarding the applicability to practice. First, the considered planning horizon of five hours is shorter than a typical workday of an LSP. However, larger instances than those considered in this computational study are too complex to be solved with an off-the-shelf solver and require more sophisticated solution methods. Second, the tactical plan obtained as a solution to the SNDMAF does not determine a definite operational solution but rather allows for different adaptations in the operational level. Third, we consider three prototypical types of heterogeneous infrastructure to cover a wide range of possible cases. In a real-world network, however, the heterogeneous infrastructure outlined by AV arcs may not fall into one of these types but may rather combine elements of each type.

We aim to address these limitations in the remainder of this thesis in the following ways. First, we propose a solution method focusing on the refinement of the underlying time-expanded networks in the subsequent Chapter 6. The objective of this customized solution method is to achieve high-quality SNDMAF solutions to real-world sized instances in reduced runtime. Then, we propose a post-processing technique in Chapter 7 for deriving an appropriate operational solution from a tactical plan. Finally, we consider a case study in Chapter 8 to assess the impact of a mixed autonomous fleet in a heterogeneous infrastructure that exists in a similar form in the real world.

Chapter 6

Dynamic discretization discovery

In this chapter, we propose and evaluate customized solution approaches for the SNDMAF. Previous experiments in Chapter 5 have shown that an off-the-shelf solver struggles to solve larger instances with reasonable efficiency. The results also indicate that solving SNDMAF instances requires on average more than eight times the runtime than solving MV only instances, which resemble conventional SND instances.

In particular, the size of the underlying time-expanded network heavily influences the complexity of an SND problem (Boland et al., 2019). The discretization, which denotes the length of the time intervals, is typically used as a parameter to adjust the accuracy and size of a time-expanded network. Considering a fine discretization leads to a large number of time intervals in the time-expanded network that enable an accurate representation of reality. Consequently, this results in an integer program with a large number of variables and constraints.

The SNDMAF particularly requires small time intervals to model an appropriate number of synchronization opportunities for vehicles and to depict the rather short travel times observed in urban environments. Unfortunately, the resulting time-expanded networks can become prohibitively large because the planning horizons considered in practical applications usually comprise several hours. Although increasing the discretization could alleviate computational efforts, the resulting time-expanded network would not depict the problem in a sufficiently accurate manner. Therefore, we propose a solution method for the SNDMAF that is based on dynamic discretization discovery (DDD). The DDD scheme was introduced by Boland et al. (2017) as an exact solution method for the basic SND problem. Instead of building *fully* time-expanded networks a priori, the DDD scheme iteratively refines *partially* time-expanded networks, which only contain a fraction of the nodes and arcs.

In the remainder of this chapter, we first review literature related to the DDD scheme and its adaptation to different problems in Section 6.1. In Section 6.2, we propose the DDD-SNDMAF algorithm and describe the measures for adapting the DDD scheme to the SNDMAF in detail. In Section 6.3, we provide two enhancements to the exact algorithm that are based on relaxations of the model. We then evaluate their performance compared to the DDD-SNDMAF and a solver in Section 6.4. In Section 6.5, we propose a heuristic search space restriction to speed up the search for high-quality primal solutions. We evaluate the performance of the heuristic approach by applying it to all DDD variants in Section 6.6. Finally, we derive implications of the proposed solution methods for the SNDMAF in Section 6.7.

6.1 Literature review

A majority of the literature that uses time-expanded networks replicates the nodes and arcs of the physical network for all time intervals of the complete planning horizon. We denote the networks obtained by this full enumeration as *fully* time-expanded networks. Although pre-processing techniques are widely used to omit variables and constraints on some nodes and arcs, the time-expanded network itself is generally not altered.

More recent approaches in literature investigate the use of time-expanded networks with varying discretization, i.e., nodes and arcs are only replicated in selected time intervals. We entitle networks that are not fully enumerated as *partially* time-expanded networks. There exist different approaches to generate partially time-expanded networks for different problems. Fleischer and Skutella (2007) apply a similar concept called condensed time-expanded networks to quickest-flow-over-time problems. Concerning traveling salesman problems with time windows (TSPTW), the literature focuses on the discretization of customers' time windows. Wang and Regan (2002) develop an approach to refine the time window discretization in an iterative fashion. For this approach, the authors prove the convergence to the optimal solution value in Wang and Regan (2009). A so-called time bucket formulation for the TSPTW is proposed by Dash et al. (2012). Time windows are partitioned into buckets which are refined using linear programming techniques.

Boland et al. (2017) propose the DDD scheme as an exact solution approach for the continuous-time service network design problem (CTSNDP). The CTSNDP is the first problem combining design decisions with time windows for which partially time-expanded networks are considered. The consideration of continuous time highlights the ability of the DDD approach to neglect an a-priori determination of the time discretization, as it is common practice for SND problems. The DDD scheme for the CTSNDP according to Boland et al. (2017) is described in more detail in Section 6.2 along with the description of the adaptation to the SNDMAF.

Recent literature focuses on adapting the DDD scheme to solve different problems and on improving its performance. He et al. (2018) propose a DDD algorithm for the minimum duration time-dependent shortest path problem in which the travel times on arcs are given by piecewise linear functions. Medina et al. (2019) apply the DDD to a combined service network design and routing problem which they formulate in two variants. Hewitt (2019) considers a continuous-time load plan design problem (CTLPDP) and proposes enhancements to the DDD algorithm to solve it. As the CTLPDP is closely related to the CTSNDP, four of these enhancements are not specific to the CTLPDP. The first enhancement is a two-phase version of the DDD in which linear relaxations are solved in the first phase. The second enhancement is the incorporation of valid inequalities that are based on feasible time windows for a commodity to take an arc. Further enhancements are a symmetry-breaking procedure for refining the partially time-expanded networks and a branching rule. Vu et al. (2020) develop a DDD algorithm for the time-dependent traveling salesman problem with time windows with objective functions that minimize either the makespan or the duration. They propose valid inequalities to eliminate subtours in the lower bound solution. Two candidate primal solutions are generated in each iteration by solving IPs on two different time-expanded networks.

Further research refers to the general idea or certain components of the DDD scheme. Belieres (2019) combines a sparse graph construction heuristic based on the DDD scheme with a Benders decomposition strategy to solve a logistics service network design problem for supply chain optimization. Lagos et al. (2020) consider a continuous-time inventory routing problem and investigate components of a potential DDD algorithm that promises exact solutions to this problem. The work of Gnegel and Fügenschuh (2020) proposes an algorithm similar to the DDD for an air-travel scheduling and routing problem.

In conclusion, the recently developed DDD scheme has been applied to a number of optimization problems arising in transportation. Applying it to the SNDMAF, however, requires several modifications. The SNDMAF differentiates itself from the CTSNDP considered by Boland et al. (2017) in three main aspects as it:

1. respects asset management considerations by ensuring design balance at nodes,
2. enforces cycles such that the number of vehicles is balanced at any location between the beginning and end of the planning horizon,
3. considers a heterogeneous fleet of vehicles that require synchronization to travel in platoons.

The first two aspects are common in a number of variants of the SND problem with resource management considerations, which have been briefly reviewed in Section 2.1.4. Thus, most of the necessary modifications presented in this work can be transferred to related problems.

6.2 Adapting DDD to the SNDMAF

We describe our adaptation of the DDD scheme to solve the SNDMAF, named the DDD-SNDMAF, in this section. The DDD-SNDMAF is based on the general DDD scheme by Boland et al. (2017) that is depicted as a flowchart in Figure 6.1. Initially, a partially time-expanded network is constructed in which the length of the arcs is underestimated compared to the correct length according to the travel time defined in the physical network. This allows the initial partially time-expanded network to only contain a fraction of the nodes and arcs of the fully time-expanded network. Consequently, the respective optimization problem solved on a partially time-expanded network overestimates consolidation opportunities and, thus, it represents a relaxation. The optimal solution value of this relaxation represents a lower bound to the optimization problem solved on the fully time-expanded network.

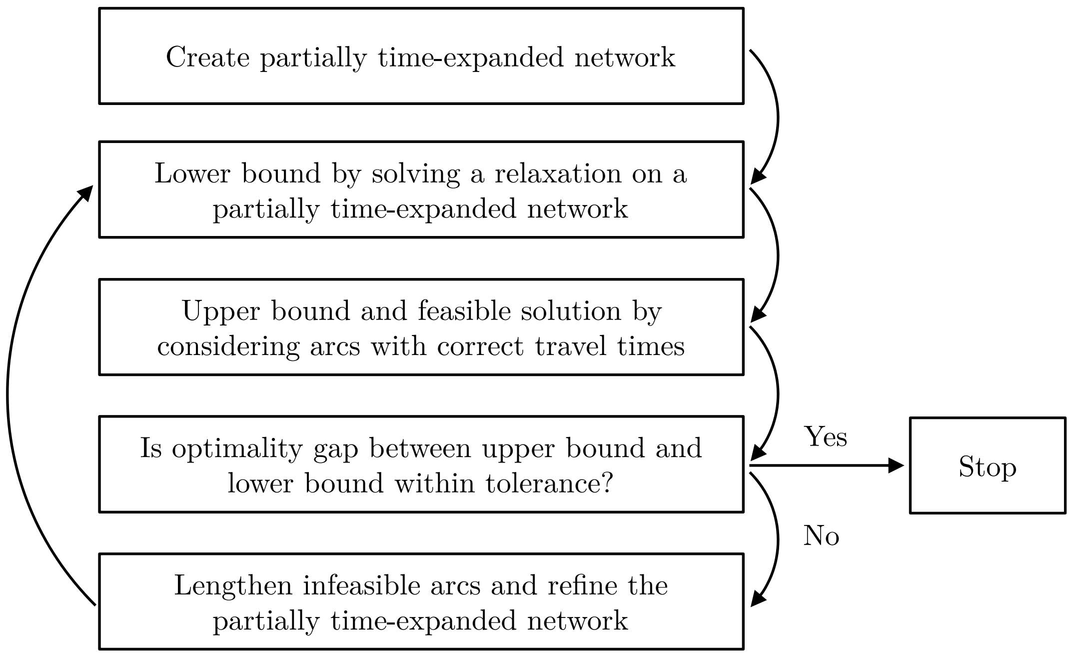


Figure 6.1: Dynamic discretization discovery scheme based on Boland et al. (2017), adapted from Scherr et al. (2020).

After obtaining a lower bound on a partially time-expanded network, a feasible solution needs to be derived to yield a corresponding upper bound for the problem. If the lower bound solution can be transferred to an upper bound solution of a value which is equal or within a given tolerance, the algorithm terminates. Else, the DDD algorithm refines the partially time-expanded network by lengthening infeasible arcs, i.e., arcs that are too short, and adding the additionally required nodes. This procedure is repeated in iterations until a partially time-expanded network is built for which the termination criterion can be achieved.

In the remainder of this section, we describe our measures to adapt this DDD scheme to the SNDMAF, thus developing the DDD-SNDMAF algorithm. Along with describing the components in detail in each section, we particularly distinguish the additions and modifications compared to the DDD algorithm introduced by Boland et al. (2017) for the CTSNDP. First, we define the properties of the partially time-expanded networks for the SNDMAF in Section 6.2.1. In Section 6.2.2, we outline the general structure of the DDD-SNDMAF algorithm. Then, the individual components of this algorithm are described in more detail in Sections 6.2.3 to 6.2.7. We refer to the DDD-SNDMAF again to derive relaxation-based enhancements in Section 6.3 and heuristic enhancements in Section 6.5.

6.2.1 Properties of the partially time-expanded networks

We define a partially time-expanded network $D_T = (N, A)$ as a directed network composed of the set of nodes N and the set of arcs A. As the partially time-expanded networks need to be constructed in a way that they provide a lower bound to the problem solved on the respective fully time-expanded network, they need to respect a particular set of properties. Boland et al. (2017) introduce four properties for the CTSNDP and prove their validity to ensure a lower bound. As these four properties can be directly applied to the SNDMAF, we describe them briefly using the notation of the SNDMAF.

PROPERTY 1 requires that D_T needs to contain the earliest origin node and the latest destination node for each commodity $k \in K$. Since the SNDMAF considers accurate time points for the departure and arrival of commodities, origin nodes $o^k = (i_o^k, t_o^k)$ and destination nodes $d^k = (i_d^k, t_d^k)$ are included in D_T.

PROPERTY 2 defines the length $\bar{t} - t$ of all arcs $((i,t),(j,\bar{t}))$ in D_T based on the respective travel time τ_{ij} defined in the physical network. Every arc in D_T needs to be at most as long as the travel time or shorter, i.e., $\bar{t} \le t + \tau_{ij}$.

PROPERTY 3 ensures that, for each node (i,t) in D_T and each arc (i,j) in the physical network that originates in the corresponding location i, there is at least one arc $((i,t),(j,\bar{t}))$ in D_T which originates in (i,t). Boland et al. (2017) note that a partially time-expanded network D_T has the so-called *early arrival* property if PROPERTIES 2 and 3 are satisfied. The DDD scheme tends to underestimate travel times in D_T because shorter arcs enable at least as much consolidation opportunities as arcs with the correct length, that are considered in the respective upper bound.

PROPERTY 4 defines that, if there exists an arc $((i,t),(j,t'))$ in D_T, there does not exist a node (j,t'') with $t' < t'' \le t + \tau_{ij}$. Thus, the arc that originates from a node is the longest of all feasible arcs with a length at most as long as the correct travel time. This property is also called the *longest-feasible arc* property.

Whereas the CTSNDP considered by Boland et al. (2017) schedules services to transport commodities from their origins to their destinations without integrating any asset management considerations, the SNDMAF seeks cyclic vehicle routes over a limited planning horizon. To consider these requirements, we define two additional properties that partially time-expanded networks for the SNDMAF need to satisfy. We introduce PROPERTIES 5 and 6 to D_T to ensure that vehicles can depart from any external zone at the beginning of the planning horizon and that they can return to any external zone at the end of the planning horizon. For this reason, we need to define feasible paths to and from the external zones, which consider the correct travel time of each arc. The additional properties are defined as follows.

PROPERTY 5 ensures that there exists at least one feasible path from each of the nodes representing an external zone at the beginning of the planning horizon, i.e., (i,t) with $i \in N_E$ and $t = t_0$, to each node (j,t') in D_T. Analogously, there needs to exist a feasible path from each node (j, t') in D_T to each node representing an external zone at the end of the planning horizon, i.e., (i, t) with $i \in N_E$ and $t = t_{max}$. These additional requirements ensure that, for any location of the physical network $i \in N_{ph}$, the earliest feasible time point t^i_{min} and the latest feasible time point t^i_{max} are considered in D_T.

PROPERTY 6 defines that every arc in D_T can be used in a feasible solution to the SNDMAF, i.e., after it is lengthened to its correct length according to the travel time. This requirement is necessary in the SNDMAF, as vehicles need to be able to return to an external zone before the end of the planning horizon t_{max}. Therefore, a lengthened arc $((i, t), (j, \bar{t}))$ in D_T must end earlier or exactly at the latest time point t^j_{max} defined for its destination location j, i.e., $t + \tau_{ij} \leq t^j_{max}$. Since we thereby eliminate arcs that would exceed the end of the planning horizon if they would be used in a feasible solution, we refer to this property as the *latest-feasible arc property.*

In the following, we visualize the differences between the partially time-expanded networks for the CTSNDP and those for the SNDMAF by means of an example. The origins and destinations of the commodities – two in this instance – are marked as filled nodes in a dedicated shade for each commodity. In Figure 6.2, we depict an initial partially time-expanded network for the CTSNDP that satisfies PROPERTIES 1 to 4 defined in Boland et al. (2017). In addition to the origin and destination nodes of the commodities, only one node in t_0 for each location is included in the initial partially time-expanded network. The end of the planning horizon is not illustrated as it is not defined in this problem.

Figure 6.3 displays an initial partially time-expanded network for the SNDMAF which – in addition to PROPERTIES 1 to 4 – satisfies PROPERTIES 5 and 6. To distinguish between the two different types of locations in the SNDMAF, nodes representing external zones are depicted as squares, i.e., in the uppermost row in this instance, and nodes representing satellites are depicted as circles. Compared to the CTSNDP, the SNDMAF additionally requires auxiliary arcs which connect the last with the first node for each external zone in the partially time-expanded network.

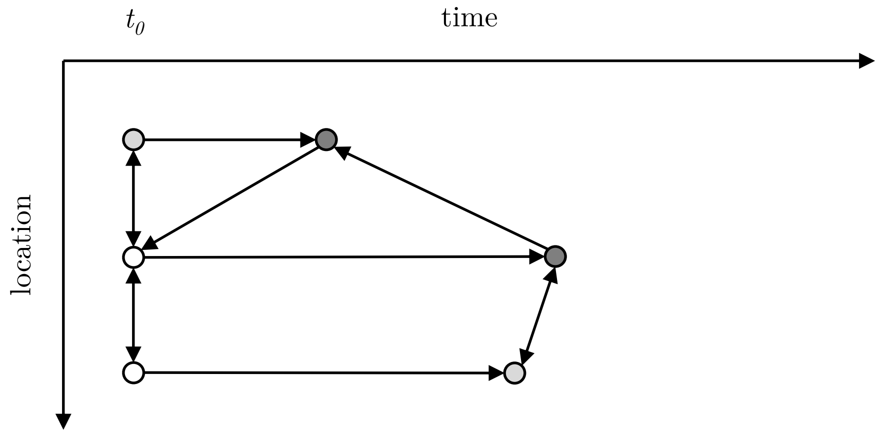


Figure 6.2: Partially time-expanded network for CTSNDP, adapted from Scherr et al. (2020).

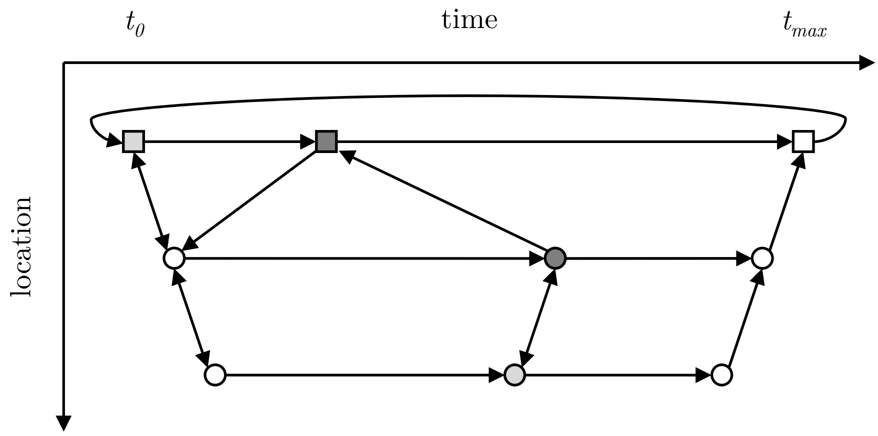


Figure 6.3: Partially time-expanded network for SNDMAF, adapted from Scherr et al. (2020).

Property 5 ensures that the width of the partially time-expanded network is limited to the length of the planning horizon. Therefore, nodes representing external zones are placed at the beginning t_0 and end t_{max} of the planning horizon. The earliest nodes for satellites are placed such that a feasible path exists from an external zone node in t_0 towards them. The latest nodes for satellites are placed such that there exists a feasible path from them towards an external zone node in t_{max}. Property 6 ensures that the arcs connecting the last nodes of each location either have the correct length according to the travel time τ_{ij} or they are not included in the partially time-expanded network.

6.2.2 DDD-SNDMAF algorithm

We depict the DDD-SNDMAF in Algorithm 1. In Iteration 0 of the DDD-SNDMAF, the initial partially time-expanded network is created according to the properties defined in the previous section using the function Create-Initial (Lines 3–4). In every iteration, the SNDMAF model is solved on the incumbent partially time-expanded network D_T to determine a lower bound $z(D_T)$ (Line 11). Then, this partially time-expanded is transformed to a corrected network D_T^τ containing only arcs of correct length, i.e., according to the realistic travel time (Line 12). On this corrected network, we also solve the SNDMAF model to obtain an upper bound $z(D_T^\tau)$ to the SNDMAF on a fully time-expanded network (Line 13).

Algorithm 1 DDD-SNDMAF.

1: iteration $= 0$
2: **while** $(z(D_T^\tau) - z(D_T))/z(D_T^\tau) > \epsilon$ **do**
3:     **if** iteration $= 0$ **then**
4:         Create-Initial
5:     **else**
6:         Identify-Nodes N_{add}
7:         **for all** $(i, t_{new}) \in N_{add}$ **do**
8:             Refine-and-Restore (i, t_{new})
9:         **end for**
10:     **end if**
11:     Solve SNDMAF(D_T) to obtain lower bound $z(D_T)$
12:     Correct D_T to D_T^τ
13:     Solve SNDMAF(D_T^τ) to obtain upper bound $z(D_T^\tau)$
14:     iteration += 1
15: **end while**

In every subsequent iteration, the partially time-expanded network is refined in the following way. Arcs that are too short and used in the incumbent lower bound solution are identified (Line 6). Then, each of these arcs is lengthened to their correct length by adding a new destination node and refining the partially time-expanded network using the function Refine-and-Restore (Lines 7–9). The updated partially time-expanded network leads to another pair of SNDMAF instances to be solved in the next iteration. The algorithm terminates if the gap based on the lower bound value $z(D_T)$ and the upper bound value $z(D_T^{\tau})$ is within a predetermined optimality tolerance ϵ. This termination criterion is enforced in Line 2. If the tolerance is set to $\epsilon = 0$, the DDD-SNDMAF terminates with a proven optimal solution to the SNDMAF. In the following sections, we describe the individual steps of the DDD-SNDMAF in more detail.

6.2.3 Creating the initial partially time-expanded network

In Iteration 0, the initial partially time-expanded network D_T is created using the function Create-Initial which we describe in Algorithm 2. The physical network D_{ph} and the commodity set K are required as input. The first nodes are created based on the origin and destination nodes of commodities k (Lines 1–4). Thus, origin nodes are created at the origin location i_o^k at the earliest time point t_o^k the commodity is available for departure. Similarly, destination nodes are created at the destination location i_d^k at the latest time point t_d^k the commodity needs to arrive.

Additional nodes are required to enable paths for each commodity from origin to destination and to allow vehicles to travel through the planning horizon starting from and ending at an external zone. Therefore, we first create nodes representing each external zone at the beginning (t_0) and end (t_{max}) of the planning horizon. Then, we create earliest and latest representatives for each satellite based on the premise that vehicles depart from an external zone in time period t_0 and return to an external zone in time period t_{max}. Thus, we create a node for each satellite at the earliest time point t_e it can be reached from any of the external zones departing in t_0. Analogously, a node is created for each satellite at the latest time point t_l from which any of the external zones can be reached until t_{max}. The time points t_e and t_l are determined based on the length of the shortest feasible path between external zones and satellites. As described, those remaining nodes are added in Lines 5–13.

Since the nodes in set N need to be connected with each other, we create arcs that are

Algorithm 2 Create-Initial.

Require: Physical network $D_{ph} = (N_{ph}, A_{ph})$, commodity set K

for all $k \in K$ **do**
    Add node (i_o^k, t_o^k) to N
    Add node (i_d^k, t_d^k) to N
end for
for all $i \in N_{ph}$ **do**
    if $i \in N_E$ **then**
        Add node (i, t_0) to N
        Add node (i, t_{max}) to N
    else
        Add node (i, t_e) to N with $t_e =$ min(length of shortest path from any $j \in N_E$ to i)
        Add node (i, t_l) to N with $t_l = t_{max} -$ min(length of shortest path from i to any $j \in N_E$)
    end if
end for
for all $(i, t) \in N$ **do**
    for all $(i, j) \in A_{ph}$ with $t + \tau_{ij} \leq t^j_{max(k)}$ **do**
        Find largest t' such that $(j, t') \in N$ and $t' \leq t + \tau_{ij}$
        Add arc $((i, t), (j, t'))$ to A_m
    end for
    Find smallest t' such that $(i, t') \in N$ and $t' > t$
    if $i \in N_E$ **then**
        Add arc $((i, t), (i, t'))$ to A_h
        if $t = t_{max}$ **then**
            Add arc $((i, t), (i, 0))$ to A_γ
        end if
    else
        Add arc $((i, t), (i, t'))$ to A_w
    end if
end for
return Partially time-expanded network $D_T = (N, A)$

defined in set A. We use the previously introduced notation of the SNDMAF and create movement arcs (A_m) to travel between physical locations, holding (A_h) and waiting arcs (A_w) to idle between subsequent time points at the same location, and auxiliary arcs (A_γ) to connect the last and first time point of each external zone. The arcs are created in Lines 14–28 based on the nodes $(i,t) \in N$ added to D_T earlier.

For movement arcs $(i,t),(j,t') \in A_m$, the destination node (j,t') is selected based on the latest time point t' in D_T that can be reached given the travel time τ_{ij} (Line 16). This is done to ensure the early arrival and the longest-feasible arc properties. To further ensure the latest-feasible arc property, movement arcs are only created if the realistic destination time $t+\tau_{ij}$, that is based on the correct travel time, does not exceed the last time point $t^j_{max(k)}$ of the destination location j that is represented in D_T (Line 15). Finally, we obtain a partially time-expanded network $D_T = (N, A)$ for the SNDMAF which respects all six properties defined earlier in Section 6.2.1.

In Figure 6.4, we provide an example for an initial partially time-expanded network created according to the described procedure. The notation table next to the partially time-expanded network depicts the origin and destination nodes of two commodities which are encoded in different shades. As external zones are depicted as squares, the origin nodes of commodities are depicted as filled squares. Destination nodes are represented as filled circles as they are satellite locations. The necessary arcs of the partially time-expanded network are depicted as black arrows.

Using dotted arrows, we show how the arcs in the partially time-expanded would look like with their correct length, i.e., after they are lengthened. Dotted squares and circles illustrate the destination nodes of these hypothetical arcs. We do not, however, illustrate the additional holding and waiting arcs these nodes would require. We can observe in the example that most of the arcs in this early stage of D_T do not have their correct length. Since multiple arcs share the same nodes as their respective origin and destination node, this allows for a smaller partially time-expanded network, particularly compared to a fully time-expanded network.

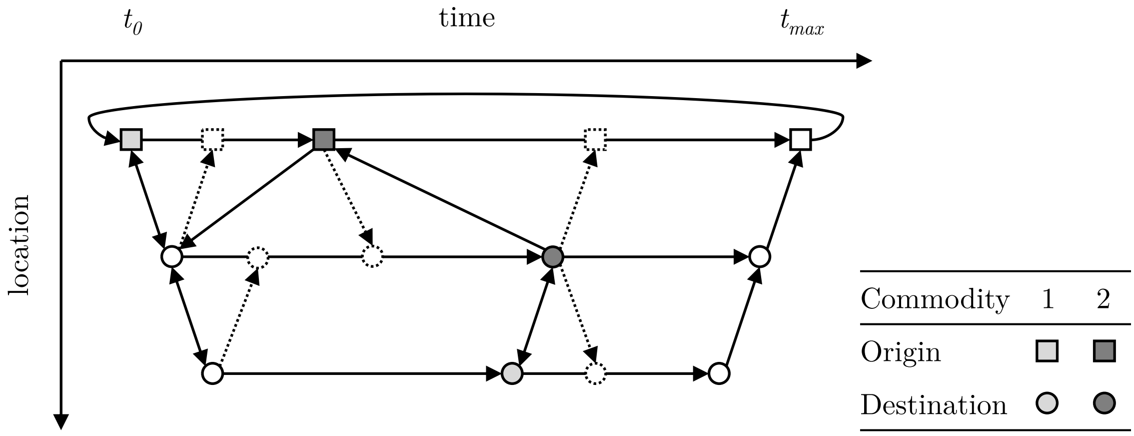


Figure 6.4: Example of an initial partially time-expanded network, adapted from Scherr et al. (2020).

6.2.4 Obtaining a lower bound

In the next step of the DDD-SNDMAF algorithm, a solution to SNDMAF(D_T), which represents a lower bound, needs to be obtained. For this, we generally use an off-the-shelf solver to solve an instance of the SNDMAF model formulated in Section 4.2. Since partially time-expanded networks, in contrast to conventional time-expanded networks, can include backward arcs, vehicles in a solution to SNDMAF(D_T) are able to travel backwards in time. In this way, the cyclic vehicle routes may not fully span the planning horizon from beginning to end and, therefore, depict unrealistic solutions. As fixed costs are determined based on the number of services installed on auxiliary arcs, which connect the end with the beginning of the planning horizon, such cycles lead to an inaccurate estimate of the fleet size and, finally, to weak lower bounds in the DDD algorithm.

In Figure 6.5, we graphically depict a solution containing this kind of cycle in the partially time-expanded network introduced earlier. The bold lines denote services which – due to the design-balance constraints – connect to a feasible cycle. Notice that, for bidirectional arcs, the thicker arrow indicates the direction of the installed service. If we assume that the vehicle performing these services is an MV, the solution value is $m_{ij}^{t\bar{t}} = 1$ on these arcs. The backward arc originating in a representative of the external zone is marked thicker to indicate that this arc is used twice in the cycle, i.e., $m_{ij}^{t\bar{t}} = 2$ on it.

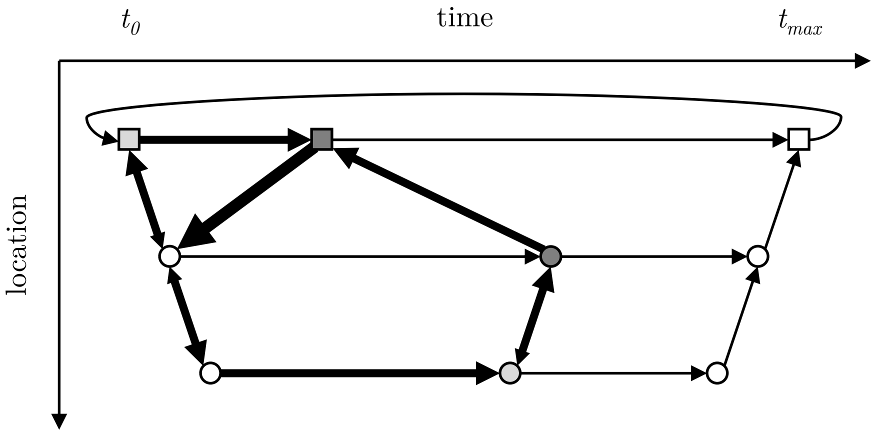


Figure 6.5: Cycle in a solution to SNDMAF(D_T) without valid inequalities, adapted from Scherr et al. (2020).

Observing the cycle, we notice that it visits all commodity origin and destination nodes. Instead of covering the complete planning horizon and using the auxiliary arc, however, the cycle uses three services on backward arcs to return to the external zone in t_0. Since the solution value on the auxiliary arc is $m_{ij}^{t\bar{t}} = 0$, the assignment of an MV, which would be required to perform the services, is not indicated. The fixed costs for the MV are also not charged such that the objective function value is underestimated compared to a feasible solution to the SNDMAF on a fully time-expanded network.

To avoid the occurrence of such cycles and to mitigate their effects, we propose valid inequalities that are considered in the SNDMAF(D_T) when obtaining a lower bound in any iteration of the DDD algorithm. The intended effects of the valid inequalities are twofold.

1. They should provide a more realistic estimate of the fleet size and mix, thus strengthening the lower bound.

2. They should "unroll" cycles such that these represent more realistic vehicle routes. This narrows the exploration of the partially time-expanded networks by reducing the number of arcs that need to be lengthened over the iterations.

We define the valid inequalities (6.1) and (6.2) in the following way. Inequalities (6.1) are applied to MV services and Inequalities (6.2) to AV services. They bound the total solution values of the service variables on auxiliary arcs on the right-hand side such that their value is larger or equal than an expected number of vehicles which is required to perform the services on movement arcs. The expected number of vehicles on the left-hand side is determined by the total travel time the services on movement arcs would require, given the correct travel times τ_{ij}, divided by the length of the planning horizon t_{max}. Services on holding and waiting arcs are not considered in this case as their length depends upon the availability of movement arcs in the incumbent partially time-expanded network. Due to the integer domain of MV and AV service variables, the solution value of the right-hand side of the valid inequalities is essentially rounded up to the next integer, which results in a stronger lower bound on the number of assigned vehicles.

$$\sum_{((i,t),(j,\bar{t}))\in A_m} \frac{\tau_{ij} m_{ij}^{t\bar{t}}}{t_{max}} \leq \sum_{((i,t),(j,\bar{t}))\in A_\gamma} m_{ij}^{t\bar{t}} \tag{6.1}$$

$$\sum_{((i,t),(j,\bar{t}))\in A_m} \frac{\tau_{ij} a_{ij}^{t\bar{t}}}{t_{max}} \leq \sum_{((i,t),(j,\bar{t}))\in A_\gamma} a_{ij}^{t\bar{t}} \tag{6.2}$$

In Figure 6.6, we illustrate a solution based on the previously introduced example that satisfies the valid inequalities. Again, arcs with a positive solution value of the service variable on them, in this case $m_{ij}^{t\bar{t}} = 1$, are depicted using bold lines. Based on the estimate that one vehicle is required to perform the services on movement arcs, the solution value on the auxiliary arc is set to $m_{ij}^{t\bar{t}} = 1$. We observe that the positive solution value on the auxiliary arc leads to a cycle that spans the complete planning horizon from t_0 to t_{max}. In contrast to the solution without valid inequalities depicted in Figure 6.5, the cycle "unrolls" to the end of the planning horizon rather than returning to t_0 without using the auxiliary arc. The obtained cycle more closely resembles a feasible vehicle route in the fully time-expanded network. Consequently, only one backward arc needs to be lengthened in the next iteration compared to the three backward arcs that are used in the solution without valid inequalities.

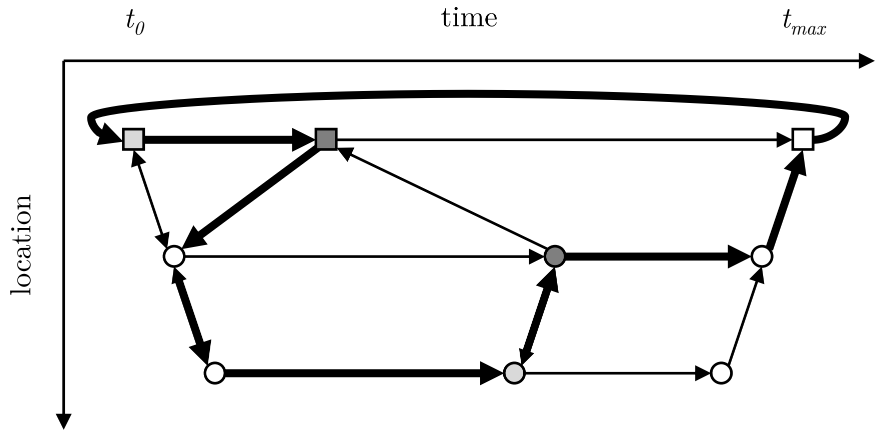


Figure 6.6: Cycle in a solution to SNDMAF(D_T) with valid inequalities, adapted from Scherr et al. (2020).

6.2.5 Identifying additional nodes

After obtaining a lower bound solution, the nodes that should be added to D_T need to be identified in the DDD-SNDMAF. The expectation is that identifying a suitable set of nodes leads to a partially time-expanded network that improves the lower bound obtained from SNDMAF(D_T). We apply the same procedure as Boland et al. (2017) to identify these nodes, which we depict in Algorithm 3. First, we define an empty set of nodes N_{add} in Line 1 to which we add nodes (i, t). The algorithm considers all movement arcs of the incumbent partially time-expanded network D_T that are shorter than they should be according to the travel time τ_{ij}, i.e., $\bar{t} - t < \tau_{ij}$ (Line 2). Then, the algorithm checks whether the respective arc is used in the solution to SNDMAF(D_T) by detecting a positive solution value of the respective service variables $m_{ij}^{t\bar{t}}$ or $a_{ij}^{t\bar{t}}$ (Line 3). If this is the case, the realistic destination node $(j, t + \tau_{ij})$, at which the movement arc ends if it would have its correct length, is added to set N_{add} in Line 4.

Referring back to the example in Figure 6.6, the only arc that is too short and used in the solution is the backward arc originating from the dark gray node representing the external zone. Using the procedure in Algorithm 3, the realistic destination node of this arc is identified as a node to be added. We graphically depict this node as a black dot in Figure 6.7, with which we illustrate the process of lengthening an arc in the subsequent

Algorithm 3 Identify-Nodes N_{add}.

Require: Partially time-expanded network $D_T = (N, A)$, solution to SNDMAF(D_T)

$N_{add} = \emptyset$
for all $((i,t),(j,\bar{t})) \in A_m$ with $\bar{t} - t < \tau_{ij}$ **do**
    if $m_{ij}^{t\bar{t}} > 0$ or $a_{ij}^{t\bar{t}} > 0$ **then**
        Add node $(j, t + \tau_{ij})$ to N_{add}
    end if
end for
return Set of nodes N_{add}

section.

6.2.6 Refining and restoring the network

In this part of the DDD-SNDMAF algorithm, the previously identified nodes in set N_{add} need to be added to the partially time-expanded network D_T. Algorithm 4 describes the procedure of adding a node $(i, t^i_{new}) \in N_{add}$ for a location i at a new time point t^i_{new}. Along with adding a node, the arcs of the partially time-expanded network need to be refined and the properties defined in Section 6.2.1 may need to be restored. In a DDD-SNDMAF iteration, this procedure is repeatedly executed for every node in N_{add} until all identified nodes are added.

Algorithm 4 works as follows. First, the node (i, t^i_{new}) is added to the set of nodes N (Line 1). A new node is always placed between existing nodes representing the same location i. Thus, the holding or waiting arc – depending on i being an external zone or a satellite – which connects the preceding node (i, t^i_k) with the succeeding node (i, t^i_{k+1}) needs to be deleted (Line 2). Then, two new holding or waiting arcs are added in Line 3, which connect the new node with the two temporally adjacent nodes.

As this new node is not connected to any movement arcs yet, we begin with adding outgoing movement arcs which originate in the new node. In Line 4, we consider the outgoing movement arcs of the preceding node at the same location, denoted as t^i_k. Then, we add copies of those arcs that arrive at the same destination node (j, t) but depart from

Algorithm 4 Refine-and-Restore (i, t^i_{new}).

Require: Node $i \in N_{ph}$ and time point $t^i_{new} \in T_i$ with $t^i_k < t^i_{new} < t^i_{k+1}$

1: Add node (i, t^i_{new}) to N
2: Delete arc $((i, t^i_k), (i, t^i_{k+1}))$ from A
3: Add arcs $((i, t^i_k), (i, t^i_{new}))$ and $((i, t^i_{new}), (i, t^i_{k+1}))$ to A
4: **for all** $((i, t^i_k), (j, t)) \in A$ with $t^i_{new} + \tau_{ij} \leq t^j_{max(k)}$ **do**
5:     Add arc $((i, t^i_{new}), (j, t))$ to A
6:     Set $t' = argmax\{s \in T_j | s \leq t^i_{new} + \tau_{ij}\}$
7:     **if** $t' \neq t$ **then**
8:         Delete arc $((i, t^i_{new}), (j, t))$ from A
9:         Add arc $((i, t^i_{new}), (j, t'))$ to A
10:     **end if**
11: **end for**
12: **for all** $((j, t), (i, t^i_k)) \in A$ such that $t + \tau_{ji} \geq t^i_{new}$ **do**
13:     Delete arc $((j, t), (i, t^i_k))$ from A
14:     Add arc $((j, t), (i, t^i_{new}))$ to A
15: **end for**

the new node at a later time point (i, t^i_{new}) in Line 5. In the condition defined in Line 4, we further exclude movement arcs that would arrive after the last representative of the destination location, i.e., $t^j_{max(k)}$, given the travel time. This is to ensure that no arc in D_T exceeds the planning horizon according to Property 6, the latest-feasible arc property.

Although we obtain the correct number of outgoing arcs, D_T may not yet satisfy Property 4 as not all of these arcs may have the longest feasible length. In Lines 6 to 10, we restore this property by identifying a node (j, t') in D_T that is at a later time point than (j, t), the current destination node of the movement arc, but can still be reached within the travel time τ_{ij}. If such a node exists, a movement arc arriving at this node replaces the incumbent, shorter movement arc which is deleted. This refine-and-restore procedure is conducted for each outgoing movement arc of the preceding node. Finally, we consider arcs that potentially arrive in the new node, i.e., incoming arcs, in Lines 12 to 15. Feasible incoming arcs for the new node are identified based on incoming arcs of the preceding node at the same location t^i_k. If there exists an arc that can reach the new, later node within the travel time, the respective arc is lengthened. In the algorithm, an arc is lengthened by deleting the previous arc and adding a new arc that arrives in the new node. Analogously to the outgoing arcs, this procedure is repeated for every incoming arc of the preceding node until the partially time-expanded network is refined.

We continue using the graphical example introduced earlier. Figure 6.7 illustrates the process of refining and restoring the partially time-expanded network after the previously identified node is added. The new node, which is depicted as a black dot, is connected via waiting arcs to the adjacent nodes. It also inherits similar outgoing and incoming arcs as the preceding node, which is the earliest representative of the respective satellite. The gray arc is lengthened in a way that it arrives at the new node. In this figure, the lengthening procedure is illustrated by a curved arrow.

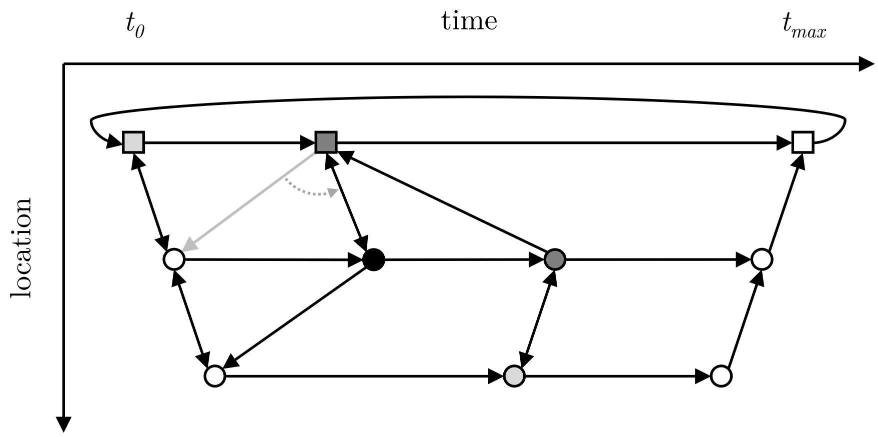


Figure 6.7: Adding a node (black dot) as well as refining and restoring the partially time-expanded network, adapted from Scherr et al. (2020).

6.2.7 Obtaining a feasible solution

The final step in an iteration of the DDD-SNDMAF is determining a feasible solution that serves as an upper bound to the optimal SNDMAF solution. Since D_T may contain arcs that are too short, a solution to SNDMAF(D_T) may not be directly transferable to a feasible solution. For this reason, we consider another kind of partially time-expanded network, denoted as the corrected network $D_T^\tau = (N^\tau, A^\tau)$, in which all arcs have their correct length according to the travel time. The corrected network should include all arcs of correct length in D_T and consider all short arcs which are used in the incumbent solution to SNDMAF(D_T) as lengthened. Nevertheless, the corrected network D_T^τ contains only a fraction of the arcs of the fully time-expanded network and even always contains fewer arcs than the partially time-expanded network D_T. Solving the integer program SNDMAF(D_T^τ) on this corrected network provides a feasible solution in any iteration of the DDD-SNDMAF algorithm.

In Algorithm 5, we describe the procedure of building a corrected network D_T^τ based on a corresponding partially time-expanded network D_T, a solution to SNDMAF(D_T), and the commodity set K. First, the origin and destination nodes of all commodities are added to node set N^τ of D_T^τ in Lines 1 to 4. Then, all auxiliary arcs are added to arc set A^τ after creating the nodes they connect, i.e., the earliest and latest nodes representing external zones (Lines 5–8). In Lines 9 to 14, the algorithm considers the movement arcs that exist in D_T. A lengthened movement arc as well as its origin node (i, t) and realistic destination node $(j, t + \tau_{ij})$ are added to D_T^τ if the corresponding arc in D_T is already lengthened or if one of the service variables $m_{ij}^{t\bar{t}}$ or $a_{ij}^{t\bar{t}}$ has a positive value in the solution to SNDMAF(D_T). Finally, holding and waiting arcs are added to the corrected network which connect all considered nodes subsequently (Lines 15–17).

Figure 6.8 illustrates a corrected network D_T^τ which is based on the incumbent partially time-expanded network D_T and the corresponding solution to SNDMAF(D_T) depicted previously in Figure 6.6. We note that the backward arc which is used in a cycle in the SNDMAF(D_T) solution is now lengthened to its correct length. The corrected network allows for determining feasible cycles to deliver all commodities in a solution to SNDMAF(D_T^τ). Since the number of available nodes and arcs is still small in this early stage of the DDD algorithm, however, performing the cycles may still require a larger number of vehicles.

Algorithm 5 Correct D_T to D_T^τ.

Require: Partially time-expanded network $D_T = (N, A)$, solution to SNDMAF(D_T), commodity set K

for all $k \in K$ **do**
  Add node (i_o^k, t_o^k) to N^τ
  Add node (i_d^k, t_d^k) to N^τ
end for
for all $((i,t),(j,\bar{t})) \in A_\gamma$ **do**
  Add nodes (i,t) and $(j,\bar{t})$ to N^τ
  Add arc $((i,t),(j,\bar{t}))$ to A^τ
end for
for all $((i,t),(j,\bar{t})) \in A_m$ **do**
  if $\bar{t} = t + \tau_{ij}$ or $m_{ij}^{t\bar{t}} > 0$ or $a_{ij}^{t\bar{t}} > 0$ **then**
    Add nodes (i,t) and $(j, t + \tau_{ij})$ to N^τ
    Add arc $((i,t),(j, t + \tau_{ij})$ to A^τ
  end if
end for
for all $(i,t) \in N^\tau$ **do**
  Add holding/waiting arcs $((i,t),(i,\bar{t}))$ to A^τ
end for
return Corrected network $D_T^\tau = (N^\tau, A^\tau)$

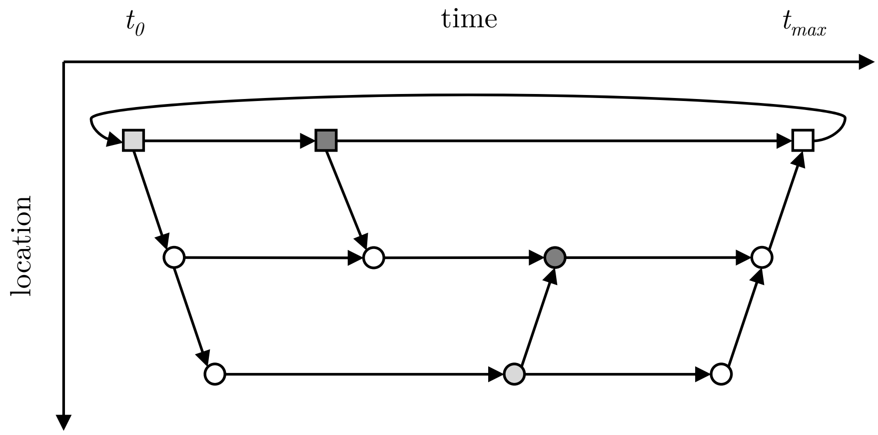


Figure 6.8: Corrected network D_T^τ.

6.3 Relaxation-based enhancements to DDD-SNDMAF

The main computational burden in the DDD-SNDMAF algorithm is solving the SNDMAF instances for computing the lower bound and the upper bound in every iteration. Although these instances are considerably smaller than those on the fully time-expanded network, solving the resulting integer programs is still challenging for the off-the-shelf solver as the SNDMAF is a special case of the capacitated multicommodity network design problem which is an $\mathcal{NP}$-hard problem (Magnanti and Wong, 1984). Decisions about installing services and routing commodities through these services with limited capacity are already of combinatorial nature. By additionally considering the design-balance constraints, feasible solutions to the integer program lie deep in the branch-and-bound tree which results in slow convergence (Pedersen et al., 2009).

To improve the performance of the DDD-SNDMAF, we propose two enhancements in this section which both consider relaxations of the SNDMAF(D_T) model. Instances of this model are solved to obtain a lower bound and to determine the arcs to be lengthened. The first enhancement is referred to as the *two-phase* DDD-SNDMAF in which the first phase relaxes all variables of the model such that they are continuous (Section 6.3.1). The second enhancement, the *partially-relaxed* DDD-SNDMAF, is a one-phase procedure in which the domains of individual variables are gradually refined from continuous to integer over the DDD iterations (Section 6.3.2).

6.3.1 Two-phase DDD-SNDMAF

The two-phase DDD-SNDMAF is based on the idea of executing the DDD in two subsequent phases. The first phase uses a relaxation of the SNDMAF model on both D_T and D_T^τ while the second phase considers the regular SNDMAF model. The two-phase scheme was first proposed by Hewitt (2019) for a load plan design problem to exploit the advantage of solving a linear program (LP) compared to an integer program (IP) in terms of the required computational effort. Since the LP relaxation permits fractional services and commodity flows, respecting the design-balance and flow conservation constraints is facilitated, and the solver can determine optimal solutions faster. Translating solutions of the LP relaxation to the corresponding IP is not trivial because simply rounding the flows may lead to infeasibility. However, the LP solutions of the first phase allow to quickly generate a partially time-expanded network with promising nodes and arcs, which serves as a starting point for the second phase. Thus, the number of iterations in the second phase, in which computationally expensive IPs need to be solved, is expected to be reduced.

We depict the scheme of the two-phase DDD-SNDMAF in Algorithm 6. The algorithm starts with a Phase 1, in which the DDD-SNDMAF (Algorithm 1) is executed. In the first phase, all variables of the SNDMAF models that are solved on D_T and D_T^τ are relaxed such that their domain is continuous. Additionally – and in contrast to Hewitt (2019) – we ignore the transportation capacity constraints (4.4) to allow for more consolidation in the first phase as the overall volume of commodities assigned to a service is unrestricted. By using the disaggregate coupling constraints (4.11) and (4.12), it is still ensured that commodities are assigned to vehicles. Although ignoring the capacity constraints underestimates the solution quality of both the SNDMAF(D_T) and the SNDMAF(D_T^τ) even further, it permits to generate a smaller partially time-expanded network at the end of Phase 1.

Algorithm 6 Two-phase DDD-SNDMAF.

1: Phase 1: Execute linear relaxation of DDD-SNDMAF with $m_{ij}^{t\bar{t}} \in \mathbb{R}$, $a_{ij}^{t\bar{t}} \in \mathbb{R}, x_{ij}^{kt\bar{t}} \in \mathbb{R}$, $y^k \in \mathbb{R}$, and without capacity constraints (4.4)
2: Phase 2: Execute DDD-SNDMAF with $m_{ij}^{t\bar{t}} \in \mathbb{N}^0$, $a_{ij}^{t\bar{t}} \in \mathbb{N}^0$, $x_{ij}^{kt\bar{t}} \in \{0,1\}$, and $y^k \in \{0,1\}$ (starting with final D_T of Phase 1)

Phase 1 ends if the termination criterion of the DDD-SNDMAF is satisfied, i.e., the objective function values of SNDMAF(D_T) and SNDMAF(D_T^τ) are within the optimality tolerance. Then, Phase 2 begins with executing another instance of the DDD-SNDMAF in which the domains of all variables are set to integer or binary and the capacity constraints are restored according to the original formulation of the SNDMAF as stated in Section 4.2. The partially time-expanded network obtained in the final iteration of Phase 1 is used in the first iteration of Phase 2 to provide a “warm start” in terms of the network. Whereas the same level of consolidation as in the first phase may not be possible in the second phase due to the capacity and integrality constraints, some of the previously generated nodes and arcs of the partially time-expanded network may also be used in an IP solution. Phase 2, similar to Phase 1, ends if the termination criterion of the DDD-SNDMAF is satisfied. Then, the two-phase DDD-SNDMAF algorithm as a whole terminates with an optimal solution to the SNDMAF.

6.3.2 Partially-relaxed DDD-SNDMAF

In the partially-relaxed DDD-SNDMAF, we also apply the idea of using a relaxation of the SNDMAF model. In contrast to the two-phase DDD-SNDMAF, however, we consider a one-phase procedure which is identical to the DDD-SNDMAF except that it considers relaxations of SNDMAF(D_T). We introduce the partially-relaxed DDD-SNDMAF in Algorithm 7. Lines 1 to 9 follow the standard DDD-SNDMAF steps that were previously presented in Algorithm 1. In Line 10 of the algorithm, a partially-relaxed SNDMAF(D_T) model is created. After considering a regular IP in Iteration 0, the partially-relaxed models in the subsequent iterations consider those variables relaxed which have not been used in any previous solution to SNDMAF(D_T).

The remaining lines of the algorithm are unaltered from the DDD-SNDMAF presented in Algorithm 1. The resulting model produces a valid lower bound in every iteration which, however, tends to be weaker than a respective lower bound obtained by an IP in the regular DDD-SNDMAF. The SNDMAF(D_T^{τ}) is never relaxed such that we obtain a feasible solution and a valid upper bound in every iteration of the partially-relaxed DDD-SNDMAF, unlike in the two-phase variant.

Before describing the procedure of creating partially-relaxed SNDMAF(D_T) models, we first define the index $r \in \{1, 2, \ldots, R\}$ for every arc $((i, t), (j, \bar{t})) \in A$ to indicate the number of DDD iterations the respective arc is considered in the partially time-expanded network. The incumbent number of overall DDD iterations is denoted as R. The index r

Algorithm 7 Partially-relaxed DDD-SNDMAF.

iteration = 0
while $(z(D_T^\tau) - z(D_T))/z(D_T^\tau) > \epsilon$ **do**
    if iteration = 0 **then**
        Create-Initial
    else
        Identify-Nodes N_{add}
        for all $(i, t_{new}) \in N_{add}$ **do**
            Refine-and-Restore (i, t_{new})
        end for
        Create partially-relaxed SNDMAF(D_T)
    end if
    Solve SNDMAF(D_T) to obtain lower bound $z(D_T)$
    Correct D_T to D_T^τ
    Solve SNDMAF(D_T^τ) to obtain upper bound $z(D_T^\tau)$
    iteration += 1
end while

only increments by one if an arc is exactly the same in subsequent iterations. Consequently, lengthened movement arcs or newly added holding and waiting arcs are assigned the index $r = 1$. We refer to this notation again in the description of the heuristic search space restriction in Section 6.5.

In Algorithm 8, we depict the procedure for creating a partially-relaxed SNDMAF(D_T) model in more detail. For every arc in the incumbent partially time-expanded network D_T (Line 1), the solution values of all variables on the respective arc are checked. Every arc is associated with service variables for MVs and AVs as well as flow variables for commodities. If the solution value of a variable has been positive in any of the previous solutions to SNDMAF(D_T), the domain of this variable is set to be integer as formulated in the SNDMAF (Lines 2–3). Otherwise, the respective variable is relaxed such that its domain is continuous (Lines 4–5). Finally, we obtain a partially-relaxed SNDMAF(D_T) model in the form of an MIP which is used to obtain a lower bound in the incumbent DDD iteration.

Algorithm 8 Create partially-relaxed SNDMAF(D_T).

Require: Partially time-expanded network $D_T = (N, A)$, solutions to SNDMAF(D_T)

for all $((i, t), (j, \bar{t})) \in A$ **do**
  if $\exists\, m_{ij}^{t\bar{t}r} > 0$ or $a_{ij}^{t\bar{t}r} > 0$ or $x_{ij}^{kt\bar{t}r} > 0$ **then**
    Set domain of variable to integer ($m_{ij}^{t\bar{t}R} \in \mathbb{N}^0$ or $a_{ij}^{t\bar{t}R} \in \mathbb{N}^0$ or $x_{ij}^{kt\bar{t}R} \in \{0, 1\}$)
  else
    Set domain of variable to continuous ($m_{ij}^{t\bar{t}R} \in \mathbb{R}$ or $a_{ij}^{t\bar{t}R} \in \mathbb{R}$ or $x_{ij}^{kt\bar{t}R} \in \mathbb{R}$)
  end if
end for
return Partially-relaxed SNDMAF(D_T)

Both the two-phase and the partially-relaxed DDD-SNDMAF always converge to the optimal solution value. The second phase of the two-phase variant is identical to the regular DDD-SNDMAF. Thus, the lower bound and the upper bound values of the IPs that are solved converge at latest when the partially time-expanded network is fully refined, i.e., all nodes and arcs of the fully time-expanded network are included. In theory, the latest point at which Phase 1 would terminate would also be the iteration in which D_T is fully refined. In this case, the optimal solution can be obtained by directly entering Phase 2 and solving the IPs in one iteration. The partially-relaxed DDD-SNDMAF, however, may still consider a relaxed version of SNDMAF(D_T) when it reaches the hypothetical iteration in which D_T is fully refined. Thus, it converges at latest in the iteration in which SNDMAF(D_T) is an IP and not relaxed anymore, i.e., all its variables have been used at least once in a previous solution.

6.4 Evaluation of the exact algorithms

In this section, we evaluate the performance of the previously introduced three variants of the DDD-SNDMAF algorithm. The evaluated algorithms are summarized and abbreviated in Table 6.1. The commercial solver *Gurobi*, which we use to solve instances of the SNDMAF model on fully time-expanded networks, serves as a benchmark to the DDD algorithms. The regular DDD-SNDMAF algorithm, which uses a one-phase procedure, is denoted as *1-DDD*. The two-phase DDD-SNDMAF is abbreviated as *2-DDD* and the partially-relaxed DDD-SNDMAF as *R-DDD*.

Table 6.1: Description of evaluated algorithms.

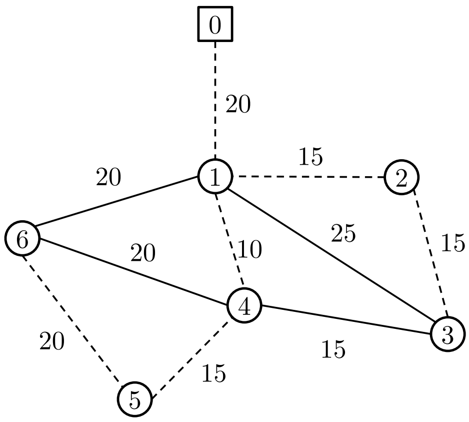
Name	Description
Gurobi	SNDMAF model solved on fully time-expanded network
1-DDD	DDD-SNDMAF
2-DDD	Two-phase DDD-SNDMAF
R-DDD	Partially-relaxed DDD-SNDMAF

The remainder of this section is structured as follows. We describe the artificially generated instances that are used as well as the experimental setup for the evaluation in Section 6.4.1. The same instances and setup are used for evaluating the heuristic algorithms later. We compare the overall performance of the algorithms in Section 6.4.2 before analyzing the individual components of 2-DDD in more detail in Section 6.4.3. Finally, we investigate and compare the quality of the obtained lower bounds in Section 6.4.4.

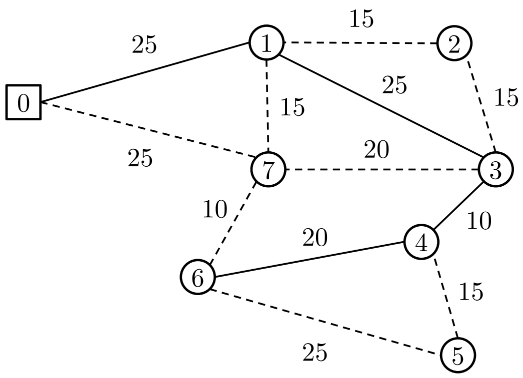
6.4.1 Instances and experimental setup

An instance is composed of a physical network, commodity demand, and the coefficients which are considered in the objective function and the constraints of the SNDMAF. We create five physical networks which are depicted in Figure 6.9. The name of each physical network is composed of the number of external zones (e) and the number of satellites (s) it contains. In the respective figures, external zones are depicted as squares, satellites as circles, MV arcs as dashed lines, and AV arcs as solid lines. Four physical networks, which are depicted in Figures 6.9a to 6.9d, comprise one external zone and a different number of satellites each, ranging from six to nine. The physical network e2_s8 depicted in Figure 6.9e comprises two external zones and eight satellites.

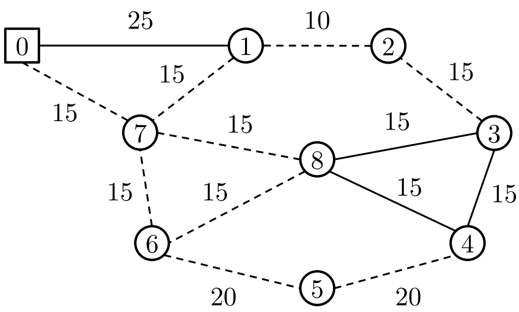
Each of the five physical networks considers four bidirectional AV arcs which are located to outline a distinct heterogeneous infrastructure. Every arc is labeled with its respective travel time as symmetric arc weights, i.e., $\tau_{ij} = \tau_{ji}$. The travel times depict realistic values



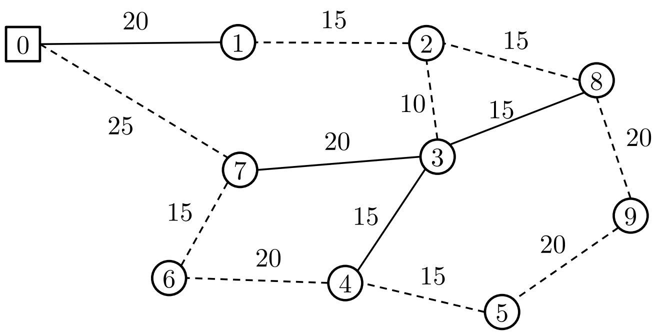
(a) e1_s6



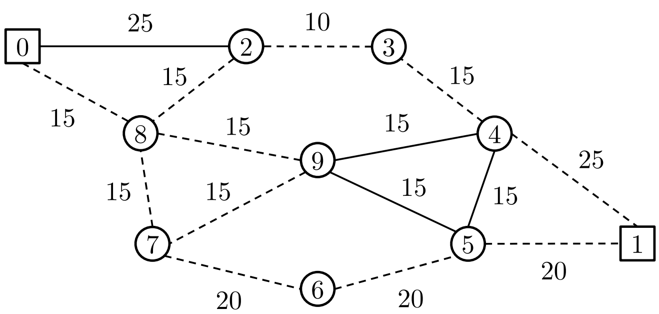
(b) e1_s7



(c) e1_s8



(d) e1_s9



(e) e2_s8

Figure 6.9: Physical networks for the evaluation of DDD-SNDMAF, adapted from Scherr et al. (2020).

for a city logistics setting with $\tau_{ij} \in \{10, 15, 20, 25\}$ in minutes. To ensure an accurate depiction of these travel times in the SNDMAF, we consider fully time-expanded networks with a discretization of $\Delta = 5$ minutes. We specify the length of the planning horizon as $t_{max} = 8$ hours based on a typical length of a daily work shift.

We consider the following demand for commodities $k \in K$ the LSP seeks to deliver from external zones to satellites. Every commodity is attributed an origin external zone i_o^k and a time point t_o^k at which it is available for departure as well as a destination satellite i_d^k and a mandatory arrival time t_d^k. At every satellite in a physical network, a number of commodities, which follows the uniform distribution $Uni(2, 4, 3)$ (min, max, mode), is scheduled to arrive over the course of the day. In this way, the number of commodities in an instance depends on the number of satellites in the respective physical network. We generate the arrival time points such that they are regularly distributed over a feasible arrival time window for each satellite. The earliest time point of this time window is determined using the shortest path from the external zones in t_0 and the latest feasible time point is determined using the shortest path to the external zones in t_{max}, respectively. Commodities arrive at a random external zone four hours ahead of their arrival time. Since the commodities cannot arrive before the planning horizon starts, some commodities are already available in t_0. Every commodity has a certain volume $v^k \sim Tri(1, 5, 3)$ which follows a triangular distribution and is defined in discrete units. The volumes of different commodities are assumed to be uncorrelated. We create ten demand instances for each physical network to yield a total of 50 instances used for the evaluation.

We define the coefficients of the SNDMAF as follows. We consider a fixed cost ratio $FCR = 2.0$ in which assigning an MV for a day costs twice as much as an AV due to the labor costs of drivers. The fixed cost of an MV is $f_M = 200$ cost units and the fixed cost of an AV is $f_A = 100$ cost units. We consider service costs of $g_{ij} = l_{ij} = 0.5$ cost units per minute of travel time, which are equal for MV and AV services. Transporting a commodity implies transportation costs of $c_{ij}^k = 0.01$ cost units per minute of travel time. The costs for outsourcing the delivery of a commodity amount to $b^k = 100$ cost units per volume unit. The driving speed of MVs and AVs is assumed to be equal such that there is no difference regarding the travel time. The transportation capacity is limited to $u_M = u_A = 20$ volume units of commodities, equally for MVs and AVs. We consider a platoon capacity of $n_{ij}^P = 2$ translating to the restriction that one MV can pull a maximum of two AVs in a platoon.

Finally, we describe the experimental setup for conducting the computational study. The algorithms are programmed using Python 3.7 and all experiments are run on an AMD Ryzen Threadripper 2990WX 32-core processor. For solving LPs and (M)IPs, we call the solver Gurobi in version 8.1.1 using the Python API (Gurobi Optimization, 2020). Gurobi uses the default settings except for the following parameters. The optimality gap of the solver is set to 1% (Gurobi parameter `MIPGap`) and the number of available threads is limited to 16 (`Threads`). For the SNDMAF(D_T) LPs and (M)IPs that are solved to obtain the lower bound, the focus of the solver is to prove optimality with a strong lower bound (`MIPFocus=2`). Then, the best known bound on the objective value (`ObjBound`) is used to update the incumbent lower bound value $z(D_T)$. The optimal solution to SNDMAF(D_T) or SNDMAF(D_T^τ) of the previous iteration is provided as a warm start to solve the respective SNDMAF model in the incumbent iteration (`NumStart`) and solutions are cut off at the upper bound value $z(D_T^\tau)$ of the previous iteration (`Cutoff`). The runtime limit for every individual solver run within an iteration of the DDD is set to 1 hour and the total runtime limit for every algorithm, including Gurobi, is set to 5 hours (`TimeLimit`). Finally, the termination criterion $(z(D_T^\tau) - z(D_T))/z(D_T^\tau) \leq \epsilon$ considers an optimality gap tolerance of $\epsilon = 1\%$ in all algorithms.

6.4.2 Overall performance

In this section, we analyze the overall performance of the exact DDD-SNDMAF algorithms and compare it to the performance of Gurobi. We depict average results over the complete set of instances here. Table 6.2 summarizes the results with each row depicting the results for one algorithm (Column "Algorithm"). We report the total runtime in seconds (Column “Runtime”) and the number of DDD iterations required (Column “#Iter.”) until the algorithm terminates. Column “$|A|$” lists the number of arcs considered in the time-expanded networks. For Gurobi, this number refers to the arcs in the fully time-expanded network. For the DDD variants, it refers to the number of arcs in the final partially time-expanded network D_T when the algorithm terminates. Since most of the variables and constraints in the SNDMAF are imposed on arcs, the magnitude of the set of arcs $|A|$ correlates with the resulting model size and, ultimately, can provide evidence on the computational complexity of an SNDMAF instance.

The solution quality produced by the different algorithms is analyzed based on the following criteria. The obtained gap at termination defined as $(z_{UB} - z_{LB})/z_{UB}$ is reported in Column “Gap”. Since the runtime limit of five hours may force a run to terminate with a sub-optimal solution, this gap can be larger than the optimality gap tolerance $\epsilon = 1\%$. We compare the average solution value z obtained using the respective DDD algorithm with that produced by Gurobi, which we define as z_{Gurobi}, in Column “Sol. qual.” by calculating $(z - z_{Gurobi})/z_{Gurobi}$. Also, we report the percentage of instances that could be solved to optimality within the tolerance $\epsilon = 1\%$. The last two columns distinguish between the number of instances that terminate with an optimal solution within the five hour runtime limit (Column “%Opt. 5h”) and those that do so within one hour of elapsed runtime (Column “%Opt. 1h”).

Table 6.2: Results for the exact algorithms, adapted from Scherr et al. (2020).

Algorithm	Runtime	#Iter.	$\|A\|$	Gap	Sol. qual.	%Opt. 5h	%Opt. 1h
Gurobi	8,412 s	–	3204.60	1.47%	–	82.00%	32.00%
1-DDD	5,163 s	14.82	725.86	1.06%	-0.24%	86.00%	62.00%
2-DDD	3,109 s	10.16	964.98	0.88%	-0.36%	90.00%	80.00%
R-DDD	4,768 s	13.66	840.30	1.06%	-0.29%	86.00%	70.00%

The results in Table 6.2 show that all three different DDD algorithms clearly surpass Gurobi in terms of both required computational effort and obtained solution quality. Overall, 1-DDD provides better solutions than Gurobi in reduced runtime and does so with the smallest partially time-expanded networks among all algorithms. 2-DDD achieves the overall best solution quality in the shortest runtime. While it requires the fewest number of iterations among the three DDD algorithms, it also generates the largest partially time-expanded networks. R-DDD strikes a balance between 1-DDD and 2-DDD in terms of the required DDD iterations and the size of the partially time-expanded networks. While the achieved solution quality of R-DDD is similar to that of 1-DDD and the runtime is shorter, it cannot achieve the same level of performance as 2-DDD.

We now compare all variants of the DDD-SNDMAF algorithm in terms of their average runtime to the solver Gurobi. To this end, Figure 6.10 depicts the runtime savings compared to the runtime required by Gurobi to solve the instances. We obtain positive savings from all variants, with 2-DDD performing best with runtime savings of 63.04%. We analyze the impact of the different components of 2-DDD on the performance in more detail in the subsequent section. While 2-DDD outperforms 1-DDD and R-DDD, both of these algorithms still terminate over 38% faster than Gurobi. The advantage from considering relaxed models of SNDMAF(D_T) is only marginal, however, as can be concluded from the relatively small runtime savings of R-DDD (43.32%) compared to 1-DDD (38.63%).

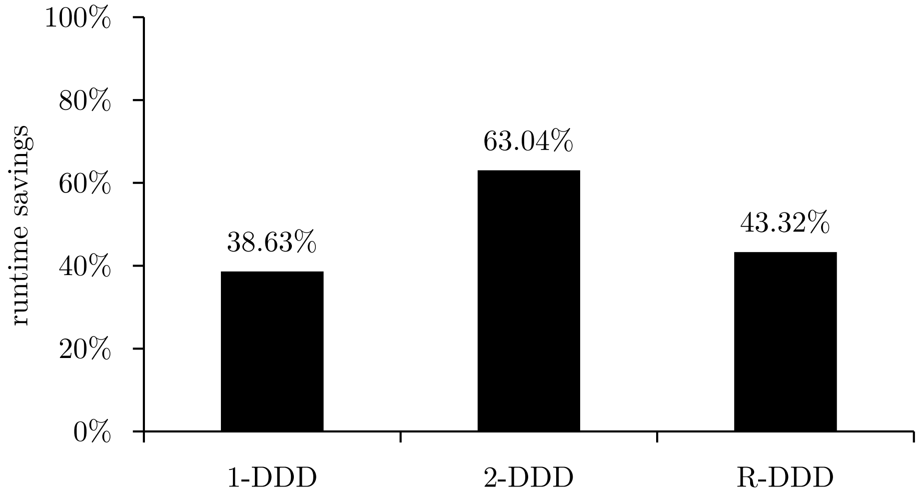


Figure 6.10: Runtime savings of DDD algorithms compared to Gurobi, adapted from Scherr et al. (2020).

Finally, we compare the number of instances that could be solved to optimality in a given time using the different algorithms. Therefore, we illustrate in Figure 6.11 the share of instances in percentage according to the two measures %Opt. 5h and %Opt. 1h introduced previously. The black bars depict the percentage of instances that could be solved to optimality within the five hour time limit (%Opt. 5h), whereas the gray bars depict the percentage of instances that terminate already within one hour of elapsed time (%Opt. 1h). Although Gurobi was able to solve 82% of the instances to optimality in five hours, it could only solve 32% in one hour. In contrast, all three DDD algorithms achieve optimal solutions to more instances than Gurobi and, in particular, were able to solve at least 62% of the instances to optimality within only one hour. We note that 80% of the instances could be solved by 2-DDD within one hour which is close to the share of instances Gurobi could solve to optimality within five hours.

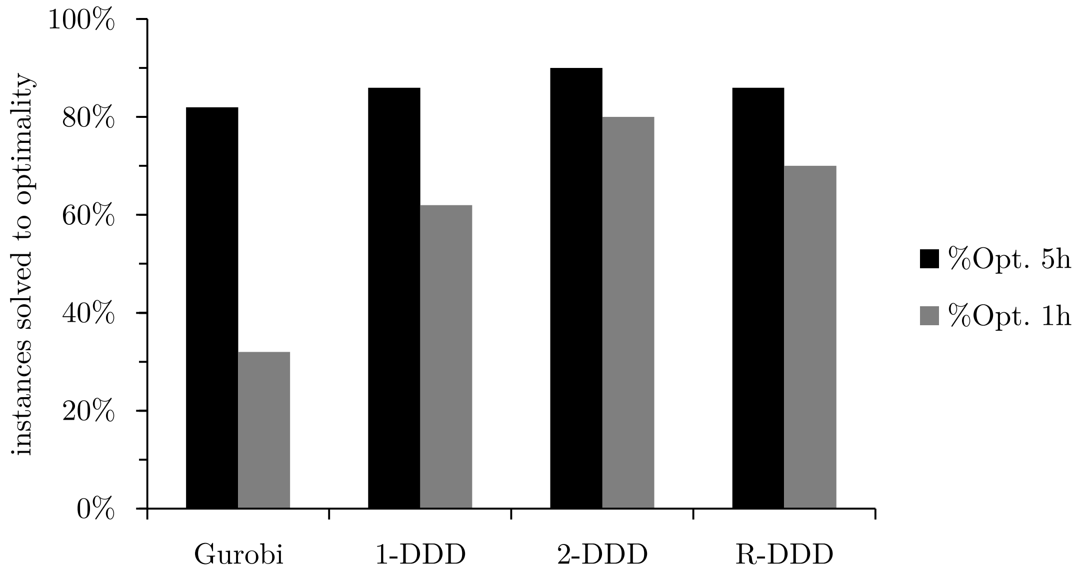


Figure 6.11: Share of instances solved to optimality within different time limits.

6.4.3 Components of two-phase DDD-SNDMAF

We further analyze the effect of the individual components of 2-DDD on its computational performance. Therefore, we consider three different 2-DDD versions of which we provide an overview in Table 6.3. Along with the name and description of the respective versions, the table contains two columns to denote the components which are included in the respective version using checkmarks. We signalize the use of valid inequalities as defined in Equations (6.1) and (6.2) in Column "VI". The consideration of the capacity constraints (4.4) in the SNDMAF models solved in Phase 1 of the algorithm is depicted in Column "Cap".

Table 6.3: Description of evaluated versions of 2-DDD regarding use of valid inequalities (VI) and consideration of capacity constraints in Phase 1 (Cap), adapted from Scherr et al. (2020).

Name	Description	VI	Cap
2-DDD	Two-phase DDD-SNDMAF	✓	
2-DDD w/o VI	2-DDD without valid inequalities		
2-DDD w/ cap	2-DDD with capacity constraints in Phase 1	✓	✓

The first version is the regular *2-DDD* as it is stated in Section 6.3.1 and evaluated in the previous section. It considers the valid inequalities – just as 1-DDD and R-DDD do – and disregards the capacity constraints in Phase 1 of the algorithm. Second, we define the version *2-DDD w/o VI* which disregards the valid inequalities and still also disregards the capacity constraints. Finally, the version *2-DDD w/ cap* considers the valid inequalities and additionally considers the capacity constraints. Based on 2-DDD, we assess the impact of the valid inequalities with version 2-DDD w/o VI and the impact of ignoring the capacity constraints in Phase 1 with version 2-DDD w/ cap.

We depict the results for the three 2-DDD versions in Table 6.4. In this table, we use the same measures we have defined previously but additionally distinguish between the two phases for every 2-DDD version in Column "Phase". In this way, Phase 1+2 denotes the overall results obtained at the end of Phase 2, i.e., the termination of the algorithm, and Phase 1 denotes the results at the end of the first phase. For Phase 1, we also measure the relative solution quality compared to Gurobi (Column "Sol. qual.") based on the final solution to SNDMAF(D_T^{τ}). Since Phase 1 considers relaxed SNDMAF models, however, this solution is not necessarily feasible, in contrast to the upper bound solutions obtained in Phase 2.

Table 6.4: Results for 2-DDD versions with different components, adapted from Scherr et al. (2020).

Algorithm	Phase	Runtime	#Iter.	$\|A\|$	Gap	Sol. qual.	%Opt. 5h
2-DDD	1+2	3,109 s	10.16	964.98	0.88%	-0.36%	90.00%
	1	90 s	6.13	773.58	–	-6.65%	–
2-DDD w/o VI	1+2	3,705 s	10.08	956.14	0.95%	-0.30%	88.00%
	1	119 s	6.48	831.28	–	-6.37%	–
2-DDD w/ cap	1+2	3,694 s	9.62	976.38	0.92%	-0.35%	90.00%
	1	122 s	5.60	849.46	–	-6.08%	–

The results show that considering the valid inequalities in 2-DDD reduces the number of iterations and the runtime until Phase 1 terminates. This is due to the more focused refinement of the partially time-expanded networks in the DDD iterations of Phase 1, which can be obtained from the relative number of arcs $|A|$ in D_T. By using the valid inequalities, the number of arcs at the end of Phase 1 is around 7% smaller than in the 2-DDD w/o VI version. The results for 2-DDD w/ cap show that considering the capacity constraints in Phase 1 reduces the number of required DDD iterations and produces a stronger lower bound at the end of Phase 1. However, the generated partially time-expanded networks at the end of Phase 1 contain around 9% more arcs. This is because the capacity constraints eliminate consolidation opportunities which, in turn, leads to a larger number of alternative vehicle paths that are considered in the SNDMAF(D_T) solutions.

Consequently, 2-DDD terminates at the end of Phase 2 around 16% faster on average than both 2-DDD w/o VI and 2-DDD w/ cap. In general, Phase 1 on average consumes only around 3% of the overall runtime. Nevertheless, it provides a partially time-expanded network that proves useful for Phase 2, as the algorithm only requires around four additional iterations to ultimately terminate.

6.4.4 Quality of the lower bounds

Although all DDD-SNDMAF algorithms provide a valid lower bound by solving an instance of SNDMAF(D_T) in every iteration, the solution quality of these lower bounds may differ because of the different kinds of relaxations that are considered. Only 1-DDD considers the SNDMAF(D_T) models as regular IPs in every iteration, whereas 2-DDD considers LPs in the first phase and R-DDD considers partially-relaxed MIPs in every iteration. Therefore, we compare the quality of the obtained lower bounds in the following. To measure the quality of the lower bound in a DDD iteration, we define the lower bound gap as $(z^* - z_{LB})/z^*$ using the incumbent lower bound value z_{LB} and the best found final solution value z^* over all algorithms.

Figure 6.12 depicts this lower bound gap for every DDD iteration as an average value over all instances. The gap is shown for all iterations up to the iteration in which the respective algorithm terminates on average. We present results for the three DDD-SNDMAF variants 1-DDD, 2-DDD, and R-DDD in this figure. In addition, we depict results for 2-DDD w/o VI to show the effect of the valid inequalities on the lower bound quality of the 2-DDD algorithm.

The results show a relatively strong convergence of the algorithms starting from the initial iteration. In all cases, the lower bound gap is smaller than 10% after only four iterations. Although 2-DDD is typically still solving LPs in Phase 1 in these first iterations, it produces lower bound gaps that, in the second iteration, even undercut those produced by the IPs solved in 1-DDD. This is due to further refined, larger partially time-expanded networks which we investigate in more detail in Section 6.6.2.

The lower bound values of 1-DDD and R-DDD are congruent in the initial iteration as R-DDD considers the SNDMAF(D_T) as an IP in this iteration. In the subsequent iterations, the lower bound values produced by R-DDD are close to those produced by 2-DDD. Since the convergence of 2-DDD slows when it nears the end of Phase 1 at around Iteration 6, the lower bound is even weaker compared to that of R-DDD. In Phase 2 of 2-DDD, the algorithm converges faster than 1-DDD and R-DDD until it terminates after

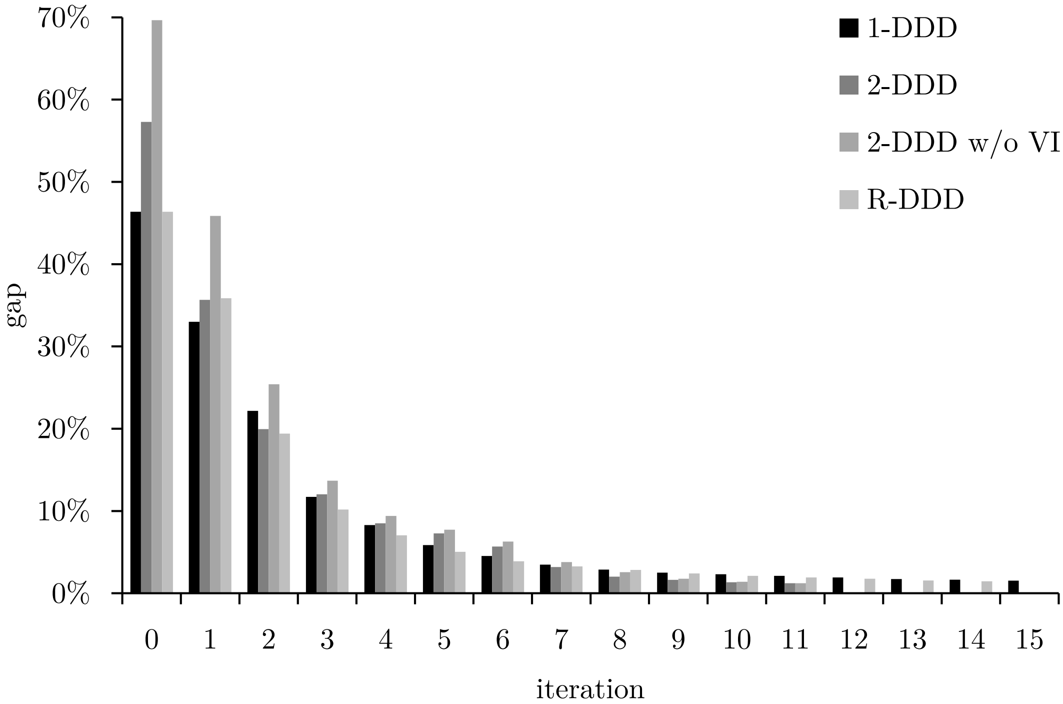


Figure 6.12: Gap of lower bound to optimal solution per iteration, adapted from Scherr et al. (2020).

the smallest number of iterations. Although the lower bound gap obtained by R-DDD is promising after only few iterations, the algorithm struggles to converge. The convergence behavior of 1-DDD is similar to that of R-DDD in the more advanced iterations.

Comparing the results for 2-DDD w/o VI with 2-DDD, we notice the overall positive effect of the valid inequalities on the quality of the lower bound. In the earlier iterations in particular, 2-DDD converges quicker, which explains its overall better performance we noted previously. Finally, we conclude that all variants of the DDD-SNDMAF algorithm converge sufficiently well and provide a faster, more reliable way to solve SNDMAF instances to optimality.

6.5 Heuristic search space restriction

In the previous sections, we have exploited ideas to find exact solutions to SNDMAF instances on partially time-expanded networks faster than an off-the-shelf-solver. In Section 6.2, we adapted the DDD algorithm to our problem and utilized knowledge about the problem to derive valid inequalities which strengthen the lower bound. In Section 6.3, we developed two enhancements, the two-phase and the partially-relaxed DDD-SNDMAF, which both rely on relaxing SNDMAF(D_T), thus exploiting knowledge about the model. The results for these exact algorithms show that a considerable number of iterations may be required to prove the optimality of a feasible solution which may have been found after relatively few iterations. The lower bound, however, can only be strengthened if either more variables are considered as integer or if the underlying partially time-expanded network is further refined. Unfortunately, both of these measures in general lead to MIPs which are harder to solve if not intractable for an off-the-shelf solver.

Therefore, we explore heuristic techniques to speed up the algorithm based on knowledge which we obtain from within the DDD as our solution approach itself. To this end, we propose the heuristic search space restriction in this section, which can be applied to all three DDD-SNDMAF variants and may be adapted to DDD algorithms for other problem settings as well. The idea is to generate heuristic cuts on MV and AV service variables in an incumbent SNDMAF(D_T) based on the solutions to SNDMAF(D_T) in the previous DDD iterations.

The objectives of the heuristic search space restriction are twofold.

1. It speeds up the process of solving the SNDMAF(D_T) in a respective iteration by providing additional, "historic" information obtained from previous iterations to the solver.

2. It provides a more narrow exploration of the partially time-expanded networks because alternative arcs, which are not part of the cycles in previous solutions, may not be used and, thus, not lengthened.

In the following, we formally introduce the heuristic cuts which are considered in SNDMAF(D_T). We define the inequalities (6.3) for MV service variables and the inequalities (6.4) for AV service variables, respectively. We refer to the notation introduced in Section 6.3.2 to denote the number of iterations an arc is included in D_T using the index $r = \{1, 2, \ldots, R\}$. Further, we define a parameter r_{min} denoting a minimum number of iterations an arc must be in D_T such that we apply cuts to the respective service variables on it. For every arc that is in D_T for more than r_{min} iterations, we enforce a lower bound on the solution value of $m_{ij}^{t\bar{t}R}$ or $a_{ij}^{t\bar{t}R}$, respectively, which is depicted on the right-hand side of the inequalities. The left-hand side defines a lower bound based on an average solution value for the respective service variable in all previous solutions to SNDMAF(D_T) since the associated arc is included in D_T. The average solution value is defined as the sum of the previous solution values divided by the number of previous iterations $R - 1$. Since service variables are defined as integer, the average solution value is rounded down.

$$\left\lfloor \frac{\sum_{r=1}^{R-1} m_{ij}^{t\bar{t}r}}{R-1} \right\rfloor \leq m_{ij}^{t\bar{t}R}, \qquad \forall((i,t),(j,\bar{t})) \in A : R > r_{min} \tag{6.3}$$

$$\left\lfloor \frac{\sum_{r=1}^{R-1} a_{ij}^{t\bar{t}r}}{R-1} \right\rfloor \leq a_{ij}^{t\bar{t}R}, \qquad \forall((i,t),(j,\bar{t})) \in A : R > r_{min} \tag{6.4}$$

By applying the heuristic search space restriction, a solution to SNDMAF(D_T) does not provide a valid lower bound to the SNDMAF anymore. Nevertheless, the termination criterion of the DDD algorithm remains the same, and the DDD terminates if a certain gap between the SNDMAF(D_T) objective value and the upper bound value obtained from SNDMAF(D_T^τ) is satisfied. Note that this gap may even be negative if the SNDMAF(D_T) objective value is larger than the incumbent upper bound value. In any case, the latest upper bound solution always provides a feasible – albeit heuristic – solution to the SNDMAF. The parameter r_{min}, however, allows to adjust the sensitivity of the heuristic search space restriction and, thereby, the impact on the solution quality.

6.6 Evaluation of the heuristic algorithms

In this second part of the evaluation, we evaluate the heuristic search space restriction introduced in the previous section. Therefore, we add the heuristic search space restriction to all three exact DDD-SNDMAF algorithms, which have been compared in the first part of the evaluation in Section 6.4. The respective heuristic algorithms, which we obtain by introducing the cuts of the heuristic search space restriction, are denoted as *1-DDD+cuts*, *2-DDD+cuts*, and *R-DDD+cuts* in the following. We conduct the experiments using the same instances and parameters as in the first part of the evaluation. We begin with assessing the effect of the heuristic search space restriction on the computational efficiency of the algorithms and the solution quality in Section 6.6.1. Then, we compare the size of the partially time-expanded networks generated by the exact and heuristic algorithms in Section 6.6.2.

6.6.1 Effect of the heuristic search space restriction

The heuristic search space restriction uses a parameter r_{min} which determines the minimum number of iterations an arc must have been included in the partially time-expanded network such that a heuristic cut is applied to a respective service variable on it. Also, r_{min} represents the denominator for calculating the average service values that are used as bounds for these cuts. In the experiments, we first set the parameter to $r_{min} = 3$, i.e., heuristic cuts in an incumbent iteration are imposed on only those arcs that have been in the partially time-expanded network in the previous three iterations. Then, we vary the value of r_{min} for 1-DDD+cuts and R-DDD+cuts to take on values of $r_{min} = 4$ and $r_{min} = 5$ to analyze the impact on runtime and solution quality. In 2-DDD+cuts, we add the heuristic cuts only in Phase 2 of the algorithm, as solving the LPs in Phase 1 is relatively inexpensive. However, the solution values of the service variables in Phase 1 are still considered for determining the lower bounds, i.e., the left-hand side, of the cuts once they are generated in Phase 2.

We report the results of these experiments in Table 6.5 based on the previously introduced measures. Additionally, we report in Column "Sol. found" the elapsed runtime at which the one primal solution has been found which is later at the termination determined the best one. Since the heuristic algorithms evaluated here do not produce valid lower bounds, we do not report the share of instances solved to optimality. Instead, we report in Column "≤Gurobi" the share of instances for which the respective algorithm found a solution of at least the same quality as Gurobi did.

Table 6.5: Results for heuristic algorithms with different values for r_{min}, adapted from Scherr et al. (2020).

Algorithm	r_{min}	Runtime	Sol. found	#Iter.	$\|A\|$	Sol. qual.	≤Gurobi
2-DDD+cuts	3	1,929 s	1,484 s	8.76	920.96	0.19%	62.00%
1-DDD+cuts	3	529 s	508 s	8.46	558.78	3.81%	10.00%
	4	958 s	786 s	10.54	612.02	1.47%	34.00%
	5	1,582 s	1,425 s	12.44	671.46	0.52%	48.00%
R-DDD+cuts	3	765 s	601 s	8.40	683.98	1.44%	26.00%
	4	1,209 s	1,022 s	9.46	739.70	0.36%	44.00%
	5	2,000 s	1,722 s	11.06	779.66	0.12%	54.00%

The results show that, compared to the exact algorithms, the heuristic search space restriction reduces the number of iterations and the size of the generated partially time-expanded networks such that the total runtime significantly decreases. The obtained solution quality with parameter setting $r_{min} = 3$ varies between the different DDD variants. 2-DDD+cuts provides solutions of similar quality compared to Gurobi, with more than half of the instances solved with the same or better solution quality. Thus, considering larger values of r_{min} to improve the solution quality is not further considered in the experiments. Both 1-DDD+cuts and R-DDD+cuts with $r_{min} = 3$, however, provide solutions of inferior quality compared to the exact algorithms and 2-DDD+cuts. Since those algorithms generate relatively small partially time-expanded networks in roughly the same number of iterations as 2-DDD+cuts requires, they terminate significantly faster. At least in the case of R-DDD+cuts, the solution quality may still be sufficient as it is only 1.44% inferior to Gurobi's on average.

To analyze whether the solution quality of 1-DDD+cuts and R-DDD+cuts can be improved, we consider the parameter values $r_{min} = 4$ and $r_{min} = 5$. In the following, we compare the performance of the algorithms using different values of r_{min} to the best performing exact algorithm, i.e., 2-DDD. In Figure 6.13, we first compare the average runtime required by the algorithms with $r_{min} = \{3, 4, 5\}$. We observe that the runtime savings compared to 2-DDD are largest for $r_{min} = 3$ and decline with an increased value of r_{min}. The largest runtime savings are achieved by 1-DDD+cuts with up to 83%. Even with $r_{min} = 5$, the savings of 1-DDD+cuts surpass those of 2-DDD+cuts with $r_{min} = 3$ clearly. The runtime savings of R-DDD+cuts show a similar sensitivity than 1-DDD+cuts, but the savings for each parameter value of r_{min} are smaller.

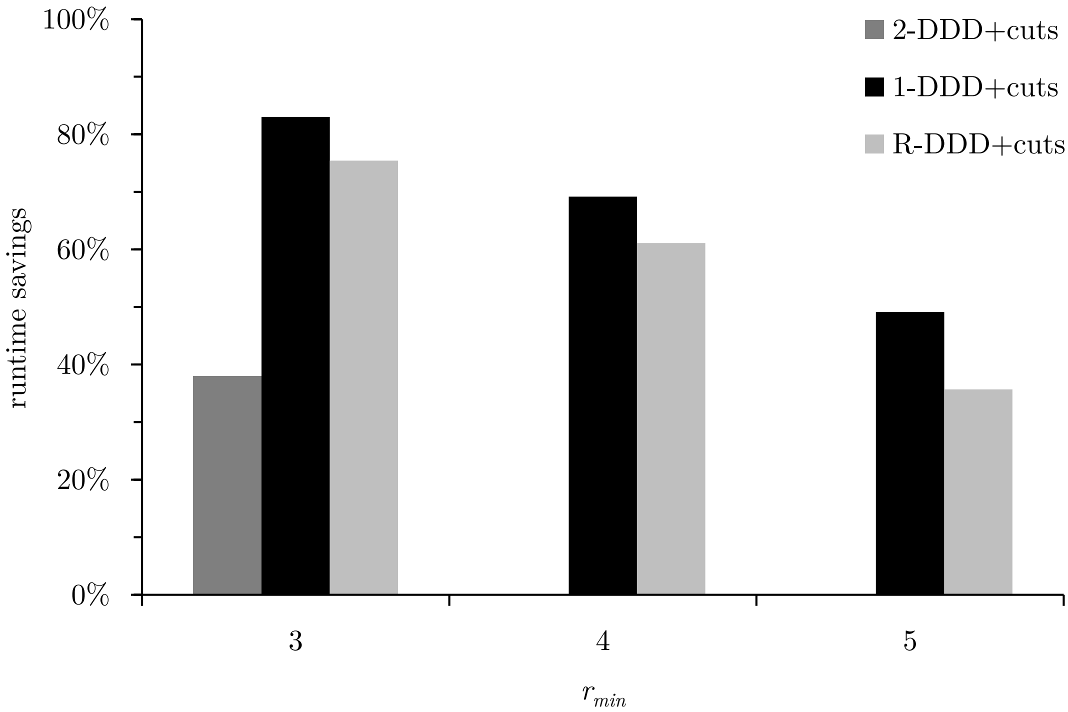


Figure 6.13: Runtime savings of heuristic algorithms compared to 2-DDD with different values for r_{min}, adapted from Scherr et al. (2020).

In Figure 6.14, we further depict the obtained relative solution quality of the previously considered heuristic algorithms compared to the solution quality of 2-DDD. The relative solution quality of a DDD algorithm is calculated as $(z - z_{2-DDD})/z_{2-DDD}$ in this case, with the objective function value of the respective algorithm denoted as z and that of 2-DDD as z_{2-DDD}. The sensitivity of the relative solution quality to the parameter r_{min} shows a similar pattern than in the figure regarding the runtime savings. In this way, setting $r_{min} = 3$ produces the weakest relative solution quality, and increasing r_{min} improves the solution quality. The solution quality of 2-DDD+cuts is close to that of 2-DDD with $r_{min} = 3$ already. The solution quality of R-DDD+cuts is also consistently better than that of 1-DDD+cuts, with a relative solution quality of under 1% for values of $r_{min} \geq 4$. In conclusion, we learn that the parameter r_{min} impacting the strength of the heuristic cuts needs to be attuned to the respective algorithm to achieve a sufficient balance between computational performance and achieved solution quality.

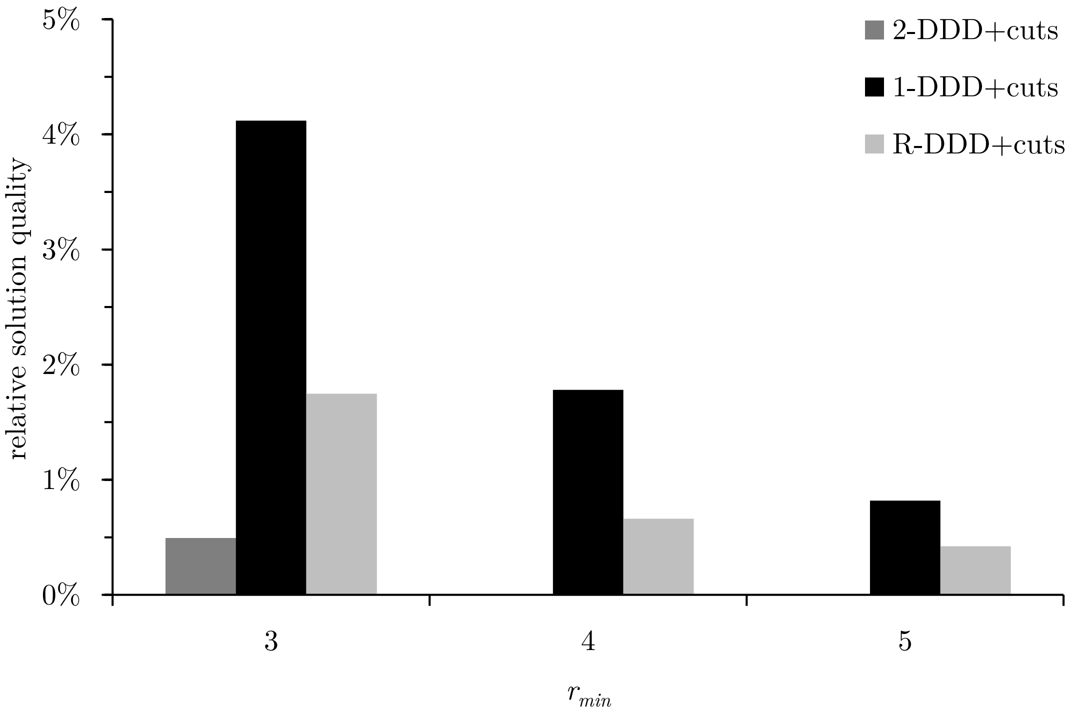


Figure 6.14: Relative solution quality of heuristic algorithms compared to 2-DDD with different values for r_{min}, adapted from Scherr et al. (2020).

6.6.2 Size of the time-expanded networks

In this section, we investigate the size of the time-expanded networks which are generated within the DDD variants. Thereby, we consider the partially time-expanded networks D_T, on which lower bounds are obtained, as well as the corrected networks D_T^τ, on which upper bounds are obtained. The size of the time-expanded networks also translates to the size of the respective SNDMAF models because most of the variables and constraints are assigned to arcs. In the following, we investigate network sizes of the three exact DDD algorithms, i.e., 1-DDD, 2-DDD, and R-DDD. We also present results for 1-DDD+cuts with $r_{min} = 3$, which produced the smallest time-expanded networks among the heuristic algorithms evaluated in the previous section (see Table 6.5).

To measure the size of the networks in a normalized way, we define the relative network size as the number of arcs in the respective network in relation to the corresponding fully time-expanded network. In this way, the number of the arcs in the fully time-expanded network, on which the complete SNDMAF model is solved using Gurobi, represents an upper bound on the number of arcs considered in D_T and D_T^τ. In the following two figures, which depict the partially time-expanded networks and the corrected networks, respectively, the relative network size is reported for every DDD iteration until the algorithm terminates on average.

In Figure 6.15, we illustrate the relative size of the partially time-expanded network D_T for the four considered DDD algorithms. The lines for the three exact algorithms are depicted in different shades, and 1-DDD+cuts as the only heuristic algorithm is depicted as a dashed line. The resulting curves feature a slope in every iteration which signalizes the arcs that are added during the refinement procedure of the iteration. From the y-intercept of the curves, we observe that the initial partially time-expanded networks, which include commodity origin and destination nodes as well as the earliest and latest nodes representing every location, contain around 5% of the arcs in the fully time-expanded network. Since arcs are never removed from D_T, the relative network size grows monotonically over the remaining iterations.

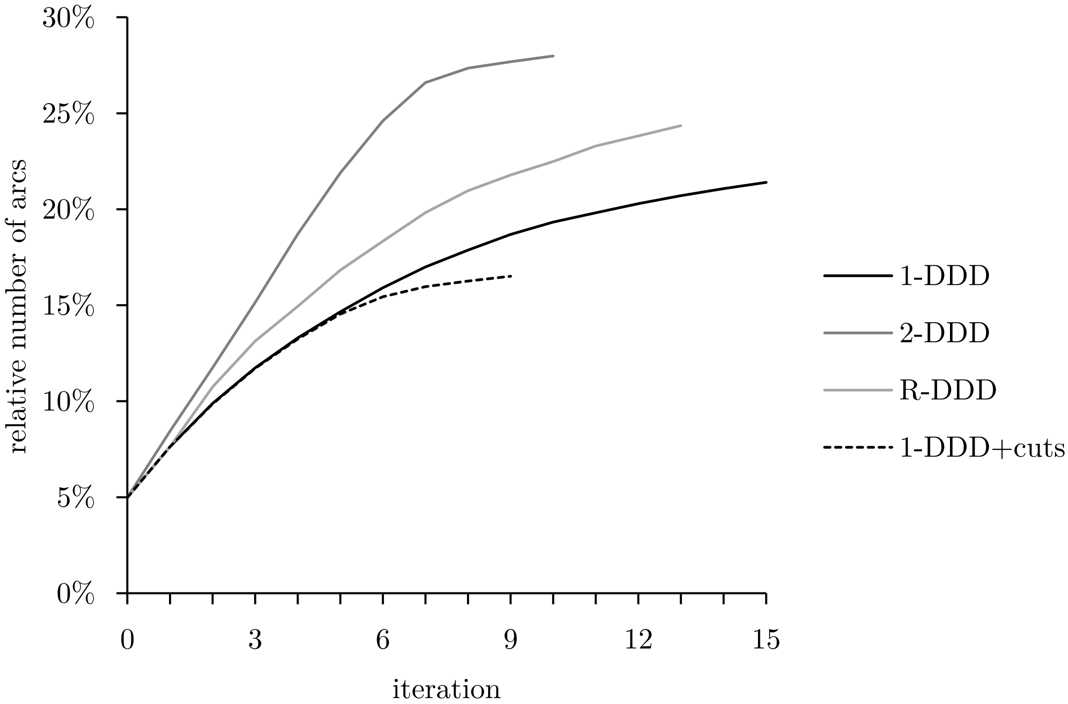


Figure 6.15: Relative size of partially time-expanded network D_T to fully time-expanded network, adapted from Scherr et al. (2020).

The slope of the curves, however, varies between the algorithms. The largest growth of D_T can be observed in Phase 1 of 2-DDD which lasts until around Iteration 6. Since the LPs that are solved in this phase allow for more flexibility, e.g., by splitting services and commodity flows over multiple paths, more arcs are used in the SNDMAF(D_T) solutions and, consequently, are lengthened. In Phase 2, the curve flattens as the exploration is limited by imposing the integrality constraints on the variables. The 1-DDD and R-DDD algorithms show a smoother growth of D_T, with 1-DDD requiring the smallest number of arcs at termination due to the consideration of IPs. The partially-relaxed MIPs in R-DDD lead to a medium growth of D_T compared to 1-DDD and 2-DDD. Introducing the heuristic search space restriction to 1-DDD decreases the growth of D_T in the iterations in which the cuts are first applied because of r_{min}. Consequently, 1-DDD+cuts terminates after fewer iterations than 1-DDD with an even smaller network size.

In Figure 6.16, we further depict the relative network size of the corrected networks D_T^τ.

Since this type of network considers the same arcs that are in D_T, albeit with their correct length, the size of D_T^τ essentially grows in proportion to D_T. Thus, the curves for the different algorithms also behave similarly. The corrected networks, however, are around one third smaller than the partially time-expanded networks. In the initial corrected network, we also notice different numbers of arcs among the algorithms. This is due to the fact that the arcs which are considered in the corrected network in an iteration, including the initial one, already take the solution to SNDMAF(D_T) in the same iteration into account. Finally, we conclude from the size of the corrected networks that 1-DDD is able to produce optimal solutions using fewer than 15% of the arcs of the fully time-expanded network. The heuristic 1-DDD+cuts even requires fewer arcs, i.e., around 10%, to produce heuristic solutions of good quality.

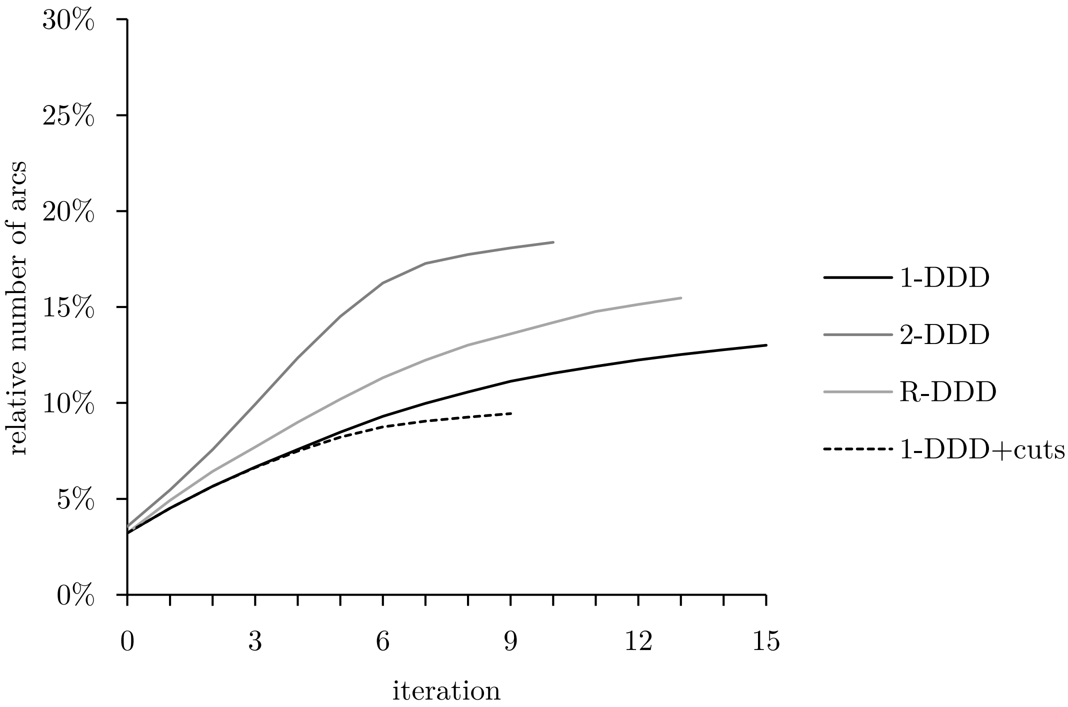


Figure 6.16: Relative size of corrected network D_T^τ to fully time-expanded network, adapted from Scherr et al. (2020).

6.7 Implications

We adapted the DDD scheme to account for design-balance constraints for cyclic vehicle routes and the mixed fleet requirements. In our variant, the DDD-SNDMAF, we consider partially time-expanded networks with two additional properties, formulate valid inequalities, and consider a corrected network on which we solve IPs to obtain an upper bound. We then enhance the DDD-SNDMAF with a two-phase variant, a partially-relaxed variant, and with proposing a heuristic search space restriction.

Particularly the two-phase DDD-SNDMAF clearly outperforms the commercial solver in solving our test instances to optimality. Its performance is also consistently superior to the other two considered exact approaches, namely the regular and the partially-relaxed DDD-SNDMAF. The heuristic search space restriction provides a valuable addition to the DDD-SNDMAF algorithm when there is the requirement for much shorter runtimes, however, this is at the expense of provable exact solutions. This balance between computational performance and solution quality can be adjusted by means of a parameter in the heuristic search space restriction that requires tuning to the respective DDD algorithm. We further conclude that applying the heuristic search space restriction to the two-phase DDD-SNDMAF provides near-optimal solutions. Combined with the regular DDD-SNDMAF, the heuristic search space restriction achieves the largest runtime savings compared to the exact algorithms.

In the following, we use the evaluated DDD-SNDMAF solution approaches to solve instances of the SNDMAF. Since the SNDMAF produces a tactical plan, SNDMAF solutions need to be transferred to an operational solution before they can be implemented. For this reason, we describe a post-processing procedure for the operationalization of a tactical plan in Chapter 7. Afterwards in Chapter 8, we consider a case study based on a realistic street network. Specifically, we solve the SNDMAF instances using the two-phase DDD-SNDMAF as well as the two-phase and the regular DDD-SNDMAF in combination with the heuristic search space restriction. Then, we apply the post-processing procedure to produce operational solutions. We analyze the tactical plans and the operational solutions obtained for this case study to derive insights into the practical application of mixed autonomous fleets in a real-world based heterogeneous infrastructure.

Chapter 7

Operationalization of a tactical plan

The tactical plan determined by the SNDMAF is based on demand forecasts and is intended to be repeatedly applied over a medium-term time horizon, e.g., a season. The objective of such a tactical plan is to ensure reliability of the LSP's services and to provide guidelines for other decision-making processes. Next to, e.g., the scheduling of workforce at external zones and satellites or the quoting of transportation capabilities in sales and marketing efforts, the implementation of a tactical plan at the operational level depicts an adjacent problem setting.

In this chapter, we thus propose a procedure to transfer a tactical plan, which allows for a certain level of flexibility, to a definite operational solution. Hereby, variations of the commodity volumes at the operational level can be incorporated. In the following, we describe this operational problem setting in Section 7.1. Thereafter, we formulate a post-processing model in Section 7.2.

7.1 Problem description

An SNDMAF solution provides a tactical plan that aims to minimize total costs. This tactical plan resembles logistics capacity planning for a medium-term time horizon. To this end, the LSP determines the overall fleet size and mix between MVs and AVs. The tactical plan further schedules a number of MVs and AVs that perform transportation services between locations at given time points and, thereby, provide the necessary load capacity to transport the expected commodity demand. For each commodity, a path from its origin to destination is determined or its delivery is outsourced. Regarding the interplay of MVs and AVs in platoons, the respective platoon lengths as well as the necessary platoon operations, i.e., split, merge, and transit operations, can be derived. In conclusion, the tactical plan minimizes the total costs associated with these activities, which are composed of fixed costs, service costs, transportation costs, and outsourcing costs.

Since the tactical plan is typically generated based on an expected commodity demand derived from historic data or forecasts, it provides some level of flexibility in terms of its implementation at the operational level. Whereas the aggregate capacities are fixed in the tactical plan, the disaggregate assignment of individual vehicles and commodities to provide those capacities can still be adjusted, e.g., on a daily basis. Along with the services and commodity paths determined in the tactical plan, the actual commodity volumes that are realized ahead of a given day may be considered. Given this level of information, the operational problem of implementing the tactical plan in the most appropriate way arises.

The decisions of the LSP associated with this operational planning problem can be divided into the following three groups.

- A feasible route needs to be derived for every individual MV and AV of the fleet. Overall, these routes need to fulfill the capacities prescribed in the tactical plan.
- The specific AVs need to be assigned to a platoon that is led by an MV. These assignments imply the required platoon operations.
- Every commodity needs to be assigned to a vehicle on every arc of its path. The

commodity assignments imply transshipments that may be necessary to transfer commodities between vehicles using loading and unloading operations.

The three groups of decisions are closely interrelated, thus we consider them in an integrated optimization problem.

The overall objective of the LSP in this operational planning problem is to achieve the consolidations prescribed in the tactical plan in the most efficient way. Therefore, we define the objective to minimize the transshipment effort because each transshipment requires handling operations for loading and unloading commodities. To achieve a minimum number of transshipments, not only the assignment of commodities to vehicles but also the assignment of vehicles to routes is crucial.

7.2 Post-processing model

In the following, we describe a post-processing model to address the problem of operationalizing an SNDMAF solution. We provide the notation in Section 7.2.1. Then, we formulate the mathematical model as an integer program in Section 7.2.2. Finally, we present an example in Section 7.2.3 to graphically depict the post-processing step.

7.2.1 Notation

We define the post-processing model for the operationalization as an IP. The input for this model is an SNDMAF solution including the underlying part of the time-expanded network that is required to depict this solution, i.e., the service network. To this end, we consider a sparse time-expanded network for the post-processing model from the arcs and nodes that are used in the solution to the SNDMAF. Therefore, we use the terms $\hat{m}_{ij}^{t\bar{t}}$ and $\hat{a}_{ij}^{t\bar{t}}$ indicating the number of MV and AV services, respectively, that are installed on an arc $((i,t),(j,\bar{t})) \in A$ in the SNDMAF solution.

To generate the underlying network for the post-processing model, we define $A^{\mathcal{M}} \subset A$ as the set of arcs with at least one MV service in the SNDMAF solution, i.e., $((i,t),(j,\bar{t})) \in$

$A^{\mathcal{M}} \leftrightarrow \hat{m}_{ij}^{t\bar{t}} \geq 1$. Analogously, $A^{\mathcal{A}} \subset A$ is defined as the set of arcs with at least one AV service in the SNDMAF solution, i.e., $((i,t),(j,\bar{t})) \in A^{\mathcal{A}} \leftrightarrow \hat{a}_{ij}^{t\bar{t}} \geq 1$. Lastly, we define the node sets $N^{\mathcal{M}}$ and $N^{\mathcal{A}}$ that are necessary to depict the respective arc sets. The resulting, sparse time-expanded network composed of the nodes $N^{\mathcal{M}} \cup N^{\mathcal{A}}$ and the arcs $A^{\mathcal{M}} \cup A^{\mathcal{A}}$ suffices for depicting the post-processing model. Naturally, this network and the resulting post-processing model are typically smaller than the corresponding fully time-expanded network and SNDMAF model.

Since the post-processing model considers each vehicle of the fleet individually, we define the set $P = \{p\}$ to denote every MV p and the set $Q = \{q\}$ to denote every AV q of the fleet. In the SNDMAF solution, the total solution value of the respective services on auxiliary arcs corresponds to the total number of assigned vehicles of a type. Thus, they define the magnitude of the MV and AV sets, with $|P| = \sum_{((i,t),(j,\bar{t})) \in A_{\gamma}^{\mathcal{M}}} \hat{m}_{ij}^{t\bar{t}}$ and $|Q| = \sum_{((i,t),(j,\bar{t})) \in A_{\gamma}^{\mathcal{A}}} \hat{a}_{ij}^{t\bar{t}}$. Based on these sets, we define the binary variable $m_{ijp}^{t\bar{t}}$ to take the value of one if the MV p travels on arc $((i,t),(j,\bar{t})) \in A^{\mathcal{M}}$, otherwise zero. Similarly, the binary variable $a_{ijq}^{t\bar{t}}$ is defined to equal one if the AV q travels on arc $((i,t),(j,\bar{t})) \in A^{\mathcal{A}}$, otherwise zero.

Further, we define a set of variables to indicate the vehicles that are traveling together in a platoon on an MV movement arc. In this way, the binary variable $\pi_{ijpq}^{t\bar{t}}$ takes the value of one if the MV p pulls the AV q via platooning on arc $((i,t),(j,\bar{t})) \in A_m^{\mathcal{A}} : (i,j) \in A_M$, otherwise zero. Similar to the notation of the SNDMAF, the platoon capacity of an MV on a physical arc (i,j) is denoted as n_{ij}^{P} and describes the maximum number of AVs in a platoon.

Assigning commodities to vehicles is another decision that the post-processing model aims to depict. Therefore, we consider the commodity paths determined in the SNDMAF solution as an input to the post-processing model. The term $\hat{x}_{ij}^{kt\bar{t}}$ indicates the flow of commodity $k \in K$ on arc $((i,t),(j,\bar{t})) \in A^{\mathcal{M}} \cup A^{\mathcal{A}}$ in the SNDMAF solution. If the delivery of a commodity k is outsourced using the variables y^k in the SNDMAF solution, this commodity can be disregarded in the post-processing model. The coefficient v^k depicts the volume in units of a commodity k. In the following, we focus on instances of the operational problem, in which the commodity demand is equal to that considered in the respective SNDMAF instance. However, the commodity demand, which is typically based on forecasts, may be updated once the actual commodity volumes are known ahead of solving the post-processing model.

The post-processing model aims to assign the positive commodity flows $\hat{x}_{ij}^{kt\bar{t}}$ to a specific vehicle on every arc traversed by commodity k on its path determined in the SNDMAF solution. Therefore, we define the binary variable $\mu_{ijkp}^{t\bar{t}}$ to take the value of one if the commodity k is assigned to the MV p on arc $((i,t),(j,\bar{t})) \in A^{\mathcal{M}}$, otherwise zero. Analogously, the binary variable $\alpha_{ijkq}^{t\bar{t}}$ equals one if the commodity k is assigned to the AV q on arc $((i,t),(j,\bar{t})) \in A^{\mathcal{A}}$, otherwise zero. Using the notation of the SNDMAF, the transportation capacity of an MV is denoted as u_M and that of an AV is denoted as u_A.

Since we assume that transshipping commodities between vehicles requires operational efforts, we measure the transshipment effort based on the number of different vehicles a commodity is assigned to on its path. To this end, we define the binary variable γ_{kp} to take the value of one if the commodity k is assigned to the MV p, otherwise zero. Similarly, the binary variable δ_{kq} equals one if the commodity k is assigned to the AV q, otherwise zero. Finally, we summarize the notation of all variables in the post-processing model in Table 7.1.

Table 7.1: Notation of the variables in the post-processing model.

Notation	Description
$m_{ijp}^{t\bar{t}} \in \{0,1\}$	MV p travels on arc $((i,t),(j,\bar{t})) \in A^{\mathcal{M}}$
$a_{ijq}^{t\bar{t}} \in \{0,1\}$	AV q travels on arc $((i,t),(j,\bar{t})) \in A^{\mathcal{A}}$
$\pi_{ijpq}^{t\bar{t}} \in \{0,1\}$	MV p pulls AV q in platoon on arc $((i,t),(j,\bar{t})) \in A_m^{\mathcal{A}} : (i,j) \in A_M$
$\mu_{ijkp}^{t\bar{t}} \in \{0,1\}$	Commodity k is assigned to MV p on arc $((i,t),(j,\bar{t})) \in A^{\mathcal{M}}$
$\alpha_{ijkq}^{t\bar{t}} \in \{0,1\}$	Commodity k is assigned to AV q on arc $((i,t),(j,\bar{t})) \in A^{\mathcal{A}}$
$\gamma_{kp} \in \{0,1\}$	Commodity k is assigned to MV p
$\delta_{kq} \in \{0,1\}$	Commodity k is assigned to AV q

7.2.2 Mathematical formulation

With the notation, the post-processing model we seek to solve is as follows:

$$\min \sum_{k\in K}\left[\sum_{p\in P}\gamma_{kp}+\sum_{q\in Q}\delta_{kq}\right] \tag{7.1}$$

subject to

$$\mu_{ijkp}^{t\bar{t}} \le \gamma_{kp}, \qquad \forall((i,t),(j,\bar{t})) \in A^{\mathcal{M}} \setminus A_h, k\in K, p\in P \tag{7.2}$$

$$\alpha_{ijkq}^{t\bar{t}} \le \delta_{kq}, \qquad \forall((i,t),(j,\bar{t})) \in A^{\mathcal{A}} \setminus A_h, k\in K, q\in Q \tag{7.3}$$

$$\sum_{p\in P}\mu_{ijkp}^{t\bar{t}}+\sum_{q\in Q}\alpha_{ijkq}^{t\bar{t}} = \hat{x}_{ij}^{kt\bar{t}}, \qquad \forall((i,t),(j,\bar{t})) \in A^{\mathcal{M}} \cup A^{\mathcal{A}} \setminus A_h, k\in K \tag{7.4}$$

$$\sum_{k\in K} v^k\mu_{ijkp}^{t\bar{t}} \le u_M m_{ijp}^{t\bar{t}}, \qquad \forall((i,t),(j,\bar{t})) \in A^{\mathcal{M}} \setminus A_h^{\mathcal{M}}, p\in P \tag{7.5}$$

$$\sum_{k\in K} v^k\alpha_{ijkq}^{t\bar{t}} \le u_A a_{ijq}^{t\bar{t}}, \qquad \forall((i,t),(j,\bar{t})) \in A^{\mathcal{A}} \setminus A_h^{\mathcal{A}}, q\in Q \tag{7.6}$$

$$a_{ijq}^{t\bar{t}} \le \sum_{p\in P}\pi_{ijpq}^{t\bar{t}}, \qquad \forall((i,t),(j,\bar{t})) \in A_m^{\mathcal{A}} : (i,j)\in A_M, q\in Q \tag{7.7}$$

$$\sum_{q\in Q}\pi_{ijpq}^{t\bar{t}} \le n_{ij}^P m_{ijp}^{t\bar{t}}, \qquad \forall((i,t),(j,\bar{t})) \in A_m^{\mathcal{A}} : (i,j)\in A_M, p\in P \tag{7.8}$$

$$\sum_{p\in P}\pi_{ijpq}^{t\bar{t}} \le 1, \qquad \forall((i,t),(j,\bar{t})) \in A_m^{\mathcal{A}} : (i,j)\in A_M, q\in Q \tag{7.9}$$

$$\sum_{((j,\bar{t}),(i,t))\in A^{\mathcal{M}}} m_{jip}^{\bar{t}t} = \sum_{((i,t),(j,\bar{t}))\in A^{\mathcal{M}}} m_{ijp}^{t\bar{t}}, \quad \forall(i,t)\in N^{\mathcal{M}} : t\neq 0, p\in P \tag{7.10}$$

$$\sum_{((j,\bar{t}),(i,t))\in A^{\mathcal{A}}} a_{jiq}^{\bar{t}t} = \sum_{((i,t),(j,\bar{t}))\in A^{\mathcal{A}}} a_{ijq}^{t\bar{t}}, \quad \forall(i,t)\in N^{\mathcal{A}} : t\neq 0, q\in Q \tag{7.11}$$

$$\sum_{((i,t),(j,\bar{t}))\in A^{\mathcal{M}}:t=0} m_{ijp}^{t\bar{t}} = 1, \qquad \forall p \in P \tag{7.12}$$

$$\sum_{((i,t),(j,\bar{t}))\in A^{\mathcal{A}}:t=0} a_{ijq}^{t\bar{t}} = 1, \qquad \forall q \in Q \tag{7.13}$$

$$\sum_{p\in P} m_{ijp}^{t\bar{t}} = \hat{m}_{ij}^{t\bar{t}}, \qquad \forall((i,t),(j,\bar{t})) \in A^{\mathcal{M}} \tag{7.14}$$

$$\sum_{q\in Q} a_{ijq}^{t\bar{t}} = \hat{a}_{ij}^{t\bar{t}}, \qquad \forall((i,t),(j,\bar{t})) \in A^{\mathcal{A}} \tag{7.15}$$

$$\gamma_{kp} \in \{0,1\}, \quad \forall k \in K, p \in P \tag{7.16}$$

$$\delta_{kq} \in \{0,1\}, \quad \forall k \in K, q \in Q \tag{7.17}$$

$$\mu_{ijkp}^{t\bar{t}} \in \{0,1\}, \quad \forall((i,t),(j,\bar{t})) \in A^{\mathcal{M}}, k \in K, p \in P \tag{7.18}$$

$$\alpha_{ijkq}^{t\bar{t}} \in \{0,1\}, \quad \forall((i,t),(j,\bar{t})) \in A^{\mathcal{A}}, k \in K, q \in Q \tag{7.19}$$

$$m_{ijp}^{t\bar{t}} \in \{0,1\}, \quad \forall((i,t),(j,\bar{t})) \in A^{\mathcal{M}}, p \in P \tag{7.20}$$

$$a_{ijq}^{t\bar{t}} \in \{0,1\}, \quad \forall((i,t),(j,\bar{t})) \in A^{\mathcal{A}}, q \in Q \tag{7.21}$$

$$\pi_{ijpq}^{t\bar{t}} \in \{0,1\}, \quad \forall((i,t),(j,\bar{t})) \in A_m^{\mathcal{A}} : (i,j) \in A_M, p \in P, q \in Q \tag{7.22}$$

The objective function (7.1) minimizes the total number of transshipments. The transshipments are measured by means of the number of vehicles, MVs and AVs, each commodity is assigned to. Since each commodity must be assigned to at least one vehicle in the operational solution, the number of shipped commodities represents a lower bound to the optimal objective value. The model is subject to the following groups of constraints. Constraints (7.2) to (7.6) model the assignment of commodities to vehicles, Constraints (7.7) to (7.9) model platooning, Constraints (7.10) to (7.15) model the creation of disaggregate vehicle routes, and Constraints (7.16) to (7.22) define the variables' domains.

In more detail, Constraints (7.2) and (7.3) enforce that a commodity must be assigned to a specific MV or AV, respectively, if the commodity is loaded on this vehicle on any arc. Constraints (7.4) ensure that every commodity flow in the SNDMAF solution is coupled with exactly one MV or AV on all arcs except for holding arcs. Constraints (7.5) and (7.6) enforce the transportation capacity restriction for every MV and for every AV, respectively.

Constraints (7.7) ensure that every AV traveling on an MV movement arc must be assigned to a platoon. Constraints (7.8) restrict the number of AVs following each MV in a platoon to its platoon capacity. We refer to these constraints as disaggregate platooning constraints in contrast to the platooning constraints of the SNDMAF, which consider the aggregate number of both MVs and AVs in platoons. Constraints (7.9) allow every AV to follow at most one MV in a platoon on an MV movement arc.

The route of every vehicle needs to fulfill the design-balance or connectivity requirement, according to which a vehicle needs to exit every node it enters. Constraints (7.10) and (7.11) enforce this requirement for MVs and AVs, respectively, at every node except for those at the beginning of the planning horizon, i.e., at every node (i,t) with $t \neq 0$. Conversely, Constraints (7.12) and (7.13) consider the nodes (i,t) with $t = 0$ and ensure that every MV and AV starts exactly one route at the beginning of the planning horizon. The distinction between those two sets of constraints is necessary to translate the cycles of the SNDMAF solution to feasible routes that cover the planning horizon by starting in $t = 0$ and ending in $t = t_{max}$. Constraints (7.14) and (7.15) couple the binary variables representing the segments of the disaggregate MV and AV routes in this post-processing model to the aggregate services in the solution to the SNDMAF. Finally, the binary domains of all variables are defined in Constraints (7.16) to (7.22).

7.2.3 Example

Using a graphical example, we intend to illustrate the operationalization of an SNDMAF solution depicting a tactical plan to yield an operational solution to the post-processing model. In Figure 7.1, we depict an excerpt of a time-expanded network representing locations i on the y-axis and time t on the x-axis. The circles depict nodes $(i,t) \in N$ and the arrows depict arcs $((i,t),(j,\bar{t})) \in A$. Two commodities are considered in this excerpt, namely commodity a and commodity b. Each commodity needs to be shipped from an origin to a destination. The origin nodes are depicted as circles in different shades of gray marking the different commodities, whereas the corresponding destination nodes are depicted as dots in the respective shade.

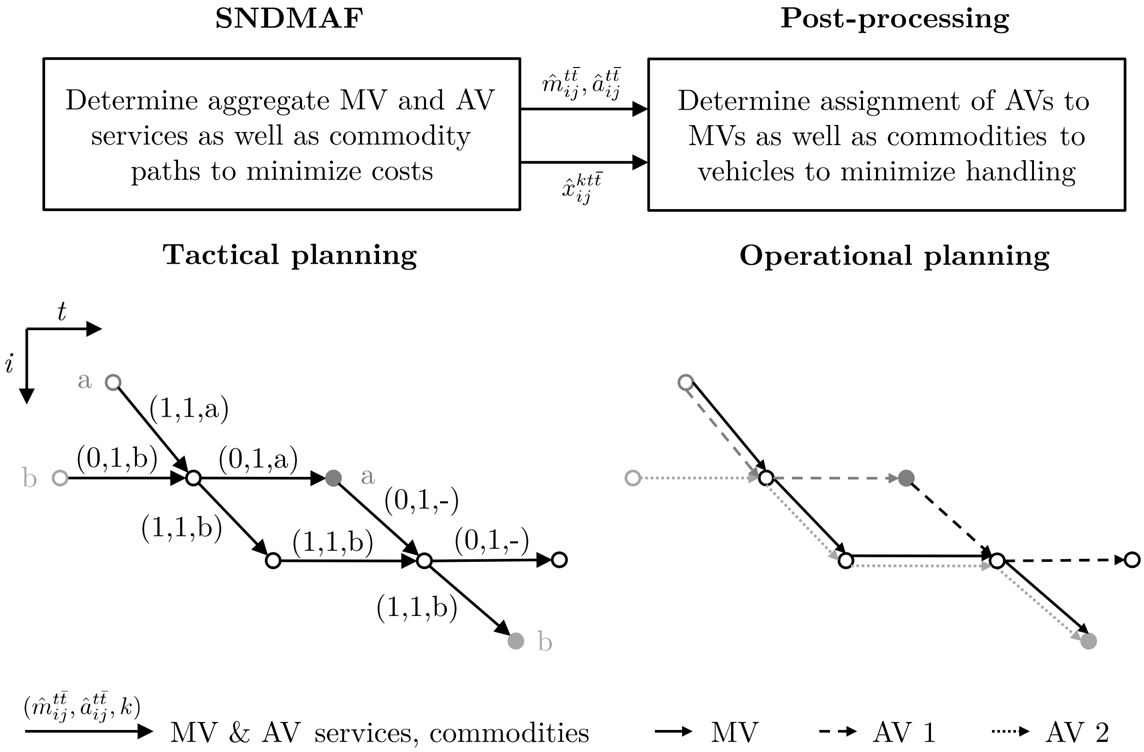


Figure 7.1: Operationalization of an SNDMAF solution using the post-processing model, adapted from Scherr et al. (2020).

The objective of the SNDMAF is to determine aggregate MV and AV services as well as commodity paths in a way that the total costs are minimized. We illustrate a possible SNDMAF solution on this part of the time-expanded network in the figure on the left-hand side. In this way, every arc $((i,t),(j,\bar{t})) \in A$ is labeled with a tuple $(\hat{m}_{ij}^{t\bar{t}}, \hat{a}_{ij}^{t\bar{t}}, k)$. This tuple denotes the number of MV services on the arc, i.e., the respective solution value of $\hat{m}_{ij}^{t\bar{t}}$, the number of AV services ($\hat{a}_{ij}^{t\bar{t}}$), and the set of commodities k with a positive commodity flow on the arc, i.e., for which $\hat{x}_{ij}^{kt\bar{t}} = 1$. In the figure, we can observe the paths of both commodities which are transported using the MV and AV services that are installed. For the sake of simplicity, we disregard the transportation capacity in this example. From the number of parallel services installed at the same time, we conclude that a total of one MV and two AVs are necessary to perform the services depicted in this example.

Eventually, the solution values regarding the installed services $\hat{m}_{ij}^{t\bar{t}}$ and $\hat{a}_{ij}^{t\bar{t}}$ as well as the solution values regarding the commodity flows $\hat{x}_{ij}^{kt\bar{t}}$ are transferred to the post-processing model for operational planning. In the operational problem, individual MV and AV routes are generated, AVs are assigned to MVs in platoons, and commodities are assigned to individual vehicles. The objective is to minimize the transshipment effort that is required when a commodity is assigned to multiple vehicles.

In the figure on the right-hand side, we depict an operational solution to the post-processing model, which is based on the SNDMAF solution presented on the left-hand side. In this example, generating a feasible route for the single MV is straightforward, as there only exists one path in the SNDMAF solution. Nevertheless, there exist multiple alternative operational solutions depicting feasible routes for the AVs. The two AVs can be assigned to a total of eight different paths, while the two commodities can each be assigned to any of the three vehicles on the arcs of their paths.

In the figure, we illustrate the optimal solution to the post-processing model in the following way. The vehicle routes of every vehicle are encoded in a different line style, with the MV depicted by solid lines and AV 1 and AV 2 depicted by dashed lines with different margins. These lines represent a positive solution value of the respective binary variable $m_{ijp}^{t\bar{t}}$ or $a_{ijq}^{t\bar{t}}$ depicting an MV p or an AV q traveling on an arc. Further, the solution specifies the AVs that follow the MV in a platoon on some arcs. In this case, the solution value to the binary variable $\pi_{ijpq}^{t\bar{t}}$ is positive if the lines of the MV p and the AV q are depicted in parallel on an arc. The lines depicting the vehicle routes are colored in the same shade of gray as the commodity that is assigned to the vehicle on the respective arc. In this way, the color encodes a positive solution value of the binary variable $\mu_{ijkp}^{t\bar{t}}$ or $\alpha_{ijkq}^{t\bar{t}}$ for a commodity k that is assigned to the MV or AV on that arc.

In the solution depicted here, commodity a is assigned to AV 1 and commodity b is assigned to AV 2 on their entire commodity paths. Since both commodities are shipped directly from their origin to destination without any transshipment necessary at intermediary nodes, this operational solution minimizes the transshipment effort and, thus, is optimal.

To emphasize the value of the post-processing model for operationalizing an SNDMAF solution, we compare the previously depicted, optimal solution with a feasible but sub-optimal solution in the following. Using Figure 7.2, we illustrate a sub-optimal operational solution using the same notation as the previous figure. Although the example is based on the same SNDMAF solution, the transshipment effort is not minimal in this sub-optimal solution. The letter T in the figure marks the nodes at which transshipments would be required to implement this operational solution. The color of the letter matches the color of the commodity that needs to be transshipped between two vehicles.

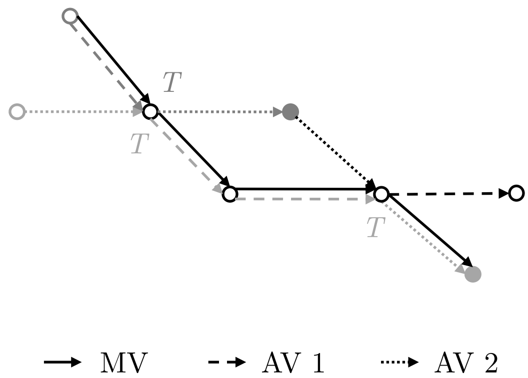


Figure 7.2: Sub-optimal operationalization of an SNDMAF solution, adapted from Scherr et al. (2020).

In this sub-optimal solution, the different selection of vehicle routes for the AVs than in the previous, optimal solution not only affects the configuration of the platoons. It also impacts the assignment of commodities to vehicles, and, ultimately, leads to an increased transshipment effort. To conclude this chapter, we note that there is value in considering the operationalization of a tactical plan, in this case an SNDMAF solution, as an optimization problem to reduce operational efforts during the implementation. In the subsequent chapter, we apply the post-processing model to the real-world sized SNDMAF instances considered in a case study to further analyze operational solutions.

Chapter 8

Case study

In the case study depicted in this chapter, we emulate the deployment of a mixed autonomous fleet in a real-world road network. To this end, we consider SNDMAF instances based on a physical network representing the city of Braunschweig, Germany. The physical network features a heterogeneous infrastructure in which the AV arcs resemble the ring road in Braunschweig, which is prepared for automated driving as part of a pilot study since 2010. The objectives of this case study are twofold. First, we evaluate the ability of select DDD-SNDMAF algorithms to solve real-world sized SNDMAF instances. Second, we investigate the structure of the tactical plans and operational solutions generated by an LSP that deploys a mixed autonomous fleet in an existing heterogeneous infrastructure.

The chapter is structured as follows. We first describe the instances and the experimental setup of the case study in Section 8.1. In Section 8.2, we present results of the study, thereby focusing on the computational performance of the applied algorithms, the impact of the mixed autonomous fleet, the usage of the vehicles in the heterogeneous infrastructure, and the operational solutions obtained from the post-processing. Finally, we derive implications of the study in Section 8.3.

8.1 Instances and experimental setup

We now describe the instances, which are composed of the physical network, the commodity demand, and the coefficients regarding costs and capacities considered in the SNDMAF model. The physical network in this study represents a potential city logistics network for Braunschweig and is illustrated in Figure 8.1. In this illustration, external zones are depicted as squares, satellites as triangles, AV arcs as solid lines, and MV arcs as dashed lines.

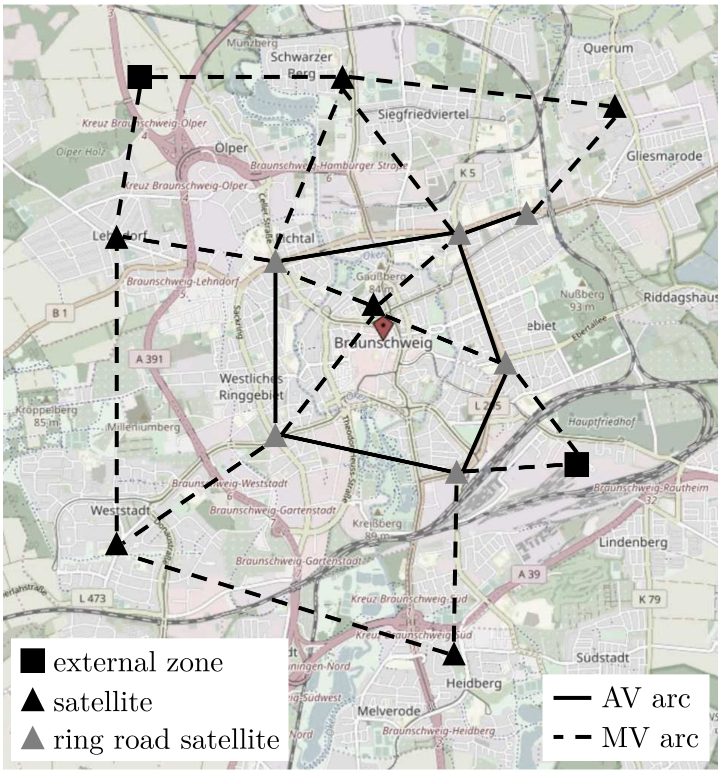


Figure 8.1: Braunschweig network, adapted from Scherr et al. (2020).

One of the two external zones is located in the northwestern corner on the city's periphery with access to the highway. The other external zone is placed at the location of DHL's city distribution center close to the central station (Persiel, 2017). The twelve satellites are distributed across the city area and are divided into two groups. Half of the satellites denoted as black triangles are located in areas of the city with a relatively high expected demand for commodities, such as residential or business districts. These satellites receive commodities that can then be further distributed in the second tier of city logistics. The other half of the satellites denoted as gray triangles are located along the city's ring road near major junctions. These six satellites do not receive commodities but rather depict intermediary locations for platoons to perform merge and split operations.

The bidirectional arcs of the physical network are distinguished into MV and AV arcs. All AV arcs are located on the city's ring road with an additional road segment on its northeastern side. As part of the project "Stadtpilot", the segments of this road are equipped for automated driving, and AVs upwards from SAE level 3 drive on it for testing purposes among regular traffic (Nothdurft et al., 2011). Recently, Ulmer and Streng (2019) also proposed to utilize AVs on this ring road in a same-day delivery problem with pickup stations. Thus, we conclude that Braunschweig represents a suitable real-world example of a heterogeneous infrastructure for our case study.

For the experiments, we consider a planning horizon with a length of $t_{max} = 8$ hours and a discretization of $\Delta = 5$ minutes for the fully time-expanded networks. Every arc is attributed a symmetric travel time in minutes, i.e., $\tau_{ij} = \tau_{ji} = \{5, 10, 15, 20\}$. We determine these travel times by obtaining average travel times from Google Maps that are rounded up to the next time interval, thus providing a conservative estimate on the actual travel time.

This physical network distinguishes itself from the types of physical networks considered in the two previous computational studies in Chapters 5 and 6 in the following ways. In contrast to the previous studies, not all satellites in the Braunschweig network receive commodities, but some are only used as meeting points for vehicles. Also, the availability of AV arcs on the ring road depicts a specific type of heterogeneous infrastructure that has not been considered previously. The heterogeneous infrastructure captures aspects of the artery type, i.e., AVs are able to travel through a large, cohesive part of the network autonomously, in this case the ring road. However, the satellites on the ring road do not receive deliveries. Also, it includes aspects of the corridor type in a way that AVs depend on MVs to travel between the external zones and the ring road in platoons.

We consider ten instances of commodity demand for the case study. The conventional, inbound demand from external zones to satellites is generated similarly to the procedure described in Section 6.4.1. In this way, every instance includes 18 inbound commodities, for each of which the origin is randomly selected between the two external zones. At each of the six satellites in the network that receive deliveries, two to four commodities are scheduled to arrive distributed over the course of the day.

Since an LSP may also collect commodities from customers at satellites, we additionally consider outbound demand in this study. Nevertheless, the majority of the freight volume in cities is typically imported. Thus, we generate nine outbound commodities such that outbound demand accounts for one third of the total demand. To this end, the origin satellite and departure time of the outbound commodities equal the destination and arrival time of every second inbound commodity. Every outbound commodity needs to arrive at the same external zone the corresponding inbound commodity departs from. The arrival times are scheduled to the end of the planning horizon t_{max}. The commodity volumes in discrete units follow the triangular distribution $v^k \sim Tri(1, 5, 3)$ and the total commodity volume is equal among the ten demand instances.

We consider the same coefficients in the SNDMAF as in Chapter 6. The fixed cost of an MV, which includes the labor costs of a driver, is $f_M = 200$ cost units, while the fixed cost of an AV is $f_A = 100$ cost units. In this way, we consider a fixed cost ratio of $FCR = 2.0$. The service costs for MV and AV services are equally set to $g_{ij} = l_{ij} = 0.5$ cost units per minute of travel time. For the transportation of a commodity, we consider the transportation costs of $c_{ij}^k = 0.01$ cost units per minute of travel time. Outsourcing the delivery of a commodity causes the outsourcing costs of $b^k = 100$ cost units per volume unit. The transportation capacity of both MVs and AVs is restricted to $u_M = u_A = 20$ volume units of commodities. Finally, we assume that a platoon led by an MV can include a maximum of two AVs, thus the platoon capacity is set to $n_{ij}^P = 2$.

For the experiments, we use exactly the same setup in terms of the used hardware, software, and parameters as in the evaluation of the DDD-SNDMAF algorithms. Therefore, we refer to the description in Section 6.4.1 for more details. Since, in this study, we also consider the implementation of the tactical plans to produce operational solutions, all instances are solved in a two-step procedure. In the first step, the SNDMAF instances are solved using Gurobi and select DDD-SNDMAF algorithms. For all algorithms, the optimality gap tolerance is set to $\epsilon = 1\%$ and the total runtime limit is set to five hours. In the second step, the resulting post-processing models are solved using Gurobi to provide optimal operational solutions based on the SNDMAF solutions obtained from the first step.

8.2 Results

We present the results to the case study in the following. First, we evaluate the computational performance of the select algorithms in solving the SNDMAF instances in Section 8.2.1. In Section 8.2.2, we assess the impact of the mixed autonomous fleet. Then, we analyze the transportation services and the use of platooning in more detail in Section 8.2.3. Finally, we investigate the structure of the operational solutions, specifically with regard to the required transshipment effort, in Section 8.2.4.

8.2.1 Computational performance

In this section, we evaluate the computational performance of three DDD-SNDMAF algorithms and Gurobi in solving the instances. Based on the results of Chapter 6, we consider the two-phase DDD-SNDMAF as the only exact DDD algorithm in this study, which we denote as *2-DDD* in the following. We further consider two heuristic algorithms by applying the heuristic search space restriction with $r_{min} = 3$ to the two-phase DDD-SNDMAF, denoted as *2-DDD+cuts*, and to the regular DDD-SNDMAF, denoted as *1-DDD+cuts*. In this way, we use the heuristic approach with the best solution quality (2-DDD+cuts) and that with the shortest runtime (1-DDD+cuts), as obtained in the results presented in Section 6.6. As a benchmark, we use the solver *Gurobi* to solve the SNDMAF instances on fully time-expanded networks.

We report the results in Table 8.1 based on the measures introduced in Chapter 6. We consider average solutions over all ten instances for every algorithm (Column "Algorithm"). We report the total runtime in seconds (Column "Runtime") and the number of DDD iterations until termination (Column "#Iter."). Also, we list the number of arcs considered in the respective final time-expanded networks in Column "$|A|$". To measure the quality of the obtained solutions, we compare the average solution value z produced by an algorithm with the solution value z_{Gurobi} produced by Gurobi. The respective values in Column "Sol. qual" are denoted as $(z - z_{Gurobi})/z_{Gurobi}$. For the exact approaches Gurobi and 2-DDD, we report the share of instances for which the optimality condition is reached in Column "%Opt.". For the DDD algorithms including the heuristic ones, we further report the share of instances for which the respective algorithm produces a solution that is at least as good as that produced by Gurobi (Column "≤Gurobi").

Table 8.1: Computational results for solving the Braunschweig instances, adapted from Scherr et al. (2020).

Algorithm	Runtime	#Iter.	$\|A\|$	Sol. qual.	%Opt.	≤Gurobi
Gurobi	16,965 s	–	5,696.00	–	10.00%	–
2-DDD	15,377 s	12.20	2,817.00	-3.02%	20.00%	80.00%
2-DDD+cuts	9,910 s	11.20	2,775.30	-3.16%	–	70.00%
1-DDD+cuts	2,587 s	12.70	1,216.50	4.99%	–	30.00%

The computational results of this case study show similar characteristics of the evaluated algorithms than the results in Chapter 6. We observe that Gurobi struggles to solve the Braunschweig instances within the five hours runtime limit and achieves an optimal solution for only one out of the ten instances. 2-DDD is able to solve the instances in around 9% shorter runtime on average. The final partially time-expanded networks are only around half of the size of the fully time-expanded network. Also, the achieved solution quality is around 3% better than Gurobi's on average. In particular, 2-DDD solves one additional instance to optimality and 80% of its solutions are at least of equal quality compared to Gurobi.

The version of 2-DDD to which the heuristic search space restriction is applied, 2-DDD+cuts, provides solutions of similar – even slightly better – quality. However, 2-DDD+cuts achieves these solutions in around 36% reduced runtime compared to 2-DDD. Since both algorithms generate final partially time-expanded networks with an almost equal number of arcs, the runtime savings are caused by the reduced number of DDD iterations and the sped up runtimes for solving the SNDMAF(D_T) models in those iterations due to the heuristic cuts. The other heuristic algorithm, 1-DDD+cuts provides solutions that are on average around 5% inferior to the solutions of Gurobi. However, the algorithm produces these solutions in less than 17% of the runtime required by Gurobi. Also, 1-DDD+cuts achieves those solutions with partially time-expanded networks that are only around 22% of the size of the fully time-expanded network.

8.2.2 Impact of the mixed autonomous fleet

We now assess the impact of introducing AVs to the fleet. To this end, we compare the fleet mix determined in the SNDMAF with an *MV only* setting which represents a conventional fleet of manually operated delivery vehicles. We depict this MV only setting by disregarding AVs in the SNDMAF model, otherwise we assume the same coefficients. The instances in the MV only setting are solved to optimality using Gurobi. Based on the results depicted in the previous section, we use the SNDMAF solutions provided by the two-phase DDD-SNDMAF for the analysis in this section and the subsequent section.

In Figure 8.2, we depict the respective fleet size and mix that is determined for each of the ten demand instances. For the SNDMAF solutions, we distinguish between the number of MVs (black bars) and AVs (gray bars), which, in total, represents the size of the mixed autonomous fleet. For the solutions in the MV only setting, we illustrate the number of MVs in each instance using a diamond symbol.

We observe that, in both the SNDMAF and MV only setting, the total fleet size ranges between three and five vehicles over all instances. The average fleet size is 3.5 vehicles in the MV only setting and 3.6 vehicles in the SNDMAF setting. The fleet size is equal

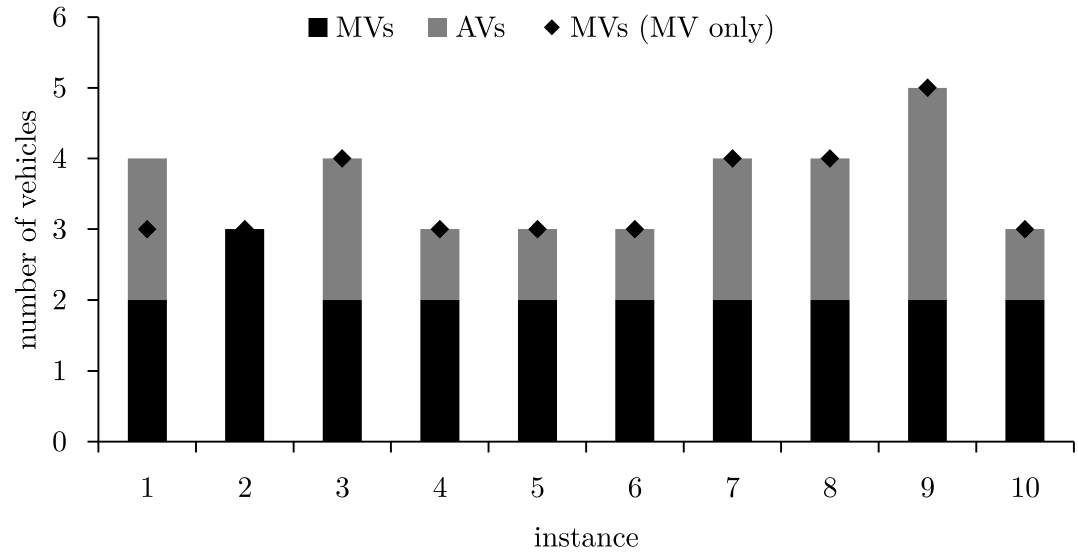


Figure 8.2: Number of MVs and AVs in mixed autonomous fleet compared to MV only setting for every demand instance.

in the SNDMAF solutions and the MV only solutions for nine out of ten instances. In the only instance, in which the mixed autonomous fleet is larger than the MV only fleet, Instance 1, the delivery of one commodity is outsourced in the MV only setting. This allows the LSP to save one MV in this setting.

The fleet mix in the SNDMAF solutions is composed of 2.1 MVs and 1.5 AVs on average. We observe that the number of assigned AVs in an individual demand instance is in general more sensitive to the total fleet size than the number of assigned MVs. Specifically, the number of MVs is equal to two in nine out of ten instances, with Instance 2 requiring three MVs. The number of AVs, however, ranges from zero to three. This leads to the conclusion that AVs are used to provide additional transportation capacity if the demand instance requires it. Overall, we conclude that the MVs in the MV only setting can be substituted to a certain extent with around the same number of AVs.

Since the fixed costs for assigning vehicles to the fleet represent a large share of the total costs, the fleet size and mix also impact the total costs. Note that we assume a fixed cost ratio of $FCR = 2.0$ in this case study, i.e., deploying an MV costs twice as much as deploying an AV. The results show that the SNDMAF solutions provide average cost savings of 6.5% for the LSP compared to the solutions in the MV only setting. In the subsequent sections, we investigate the usage of the individual vehicles of the mixed autonomous fleet in more detail.

8.2.3 Transportation services and platooning

In this section, we investigate the transportation services performed by MVs and AVs in the SNDMAF solutions to the case study. We specifically focus on the services on movement arcs for traveling between locations, which we also denote as *moves*. The travel time consumed by conducting these moves has an impact on the service costs of the LSP and, due to the usage of public infrastructure, on the city's traffic. We first obtain the MV and AV moves with their respective travel time from the SNDMAF solutions. Then, we derive the MV and AV moves in platoons using Equations (5.1) and (5.2) defined in Section 5.2.4.

In Figure 8.3, we depict the share of the planning horizon t_{max} that an MV or AV spends on performing moves with a specific total travel time. The black bars denote the travel time consumed by all moves and the gray bars denote the travel time of the moves in platoons. The remaining share of the planning horizon not used for performing moves denotes the time a vehicle spends idling at locations, i.e., for using holding and waiting arcs in the SNDMAF solution. We distinguish between the MVs and AVs of the mixed autonomous fleet. For the sake of comparison, we also consider the MVs in the MV only setting that only perform moves outside of platoons.

We observe different types of vehicle usage from the values depicted in this figure. In the SNDMAF solutions, MVs are used for performing moves the majority of the time they are available. AVs, however, are only used for 14.03% of the length of the planning horizon to travel between locations. Compared with the MV only setting, in which the MVs perform moves 30.18% of the time, the fewer MVs in the mixed autonomous fleet are each used more extensively for moves. Regarding platooning, the two vehicle types in the SNDMAF also show differences. Whereas MVs consume only 15.25% of their total

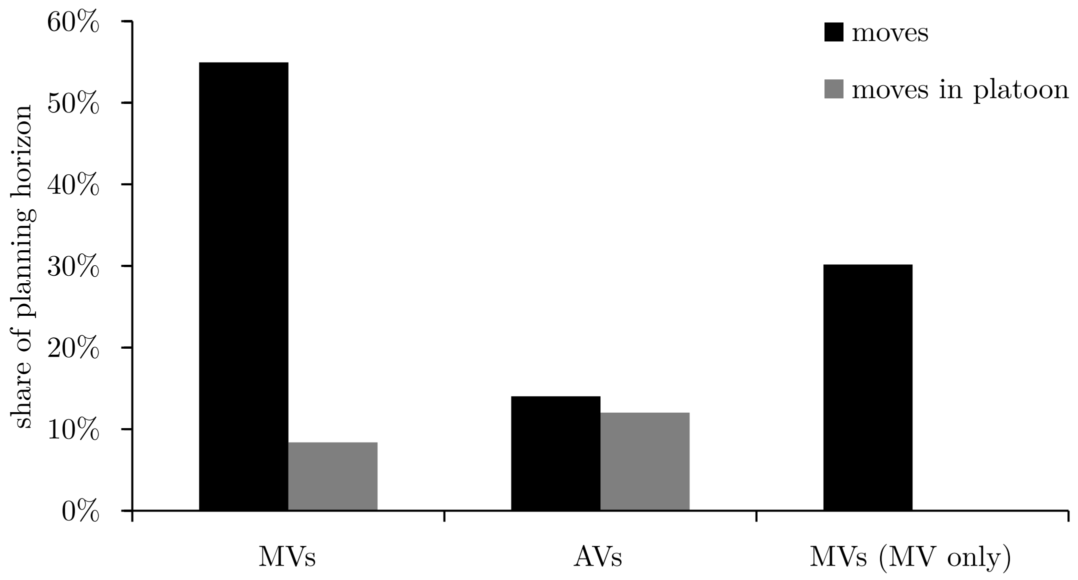
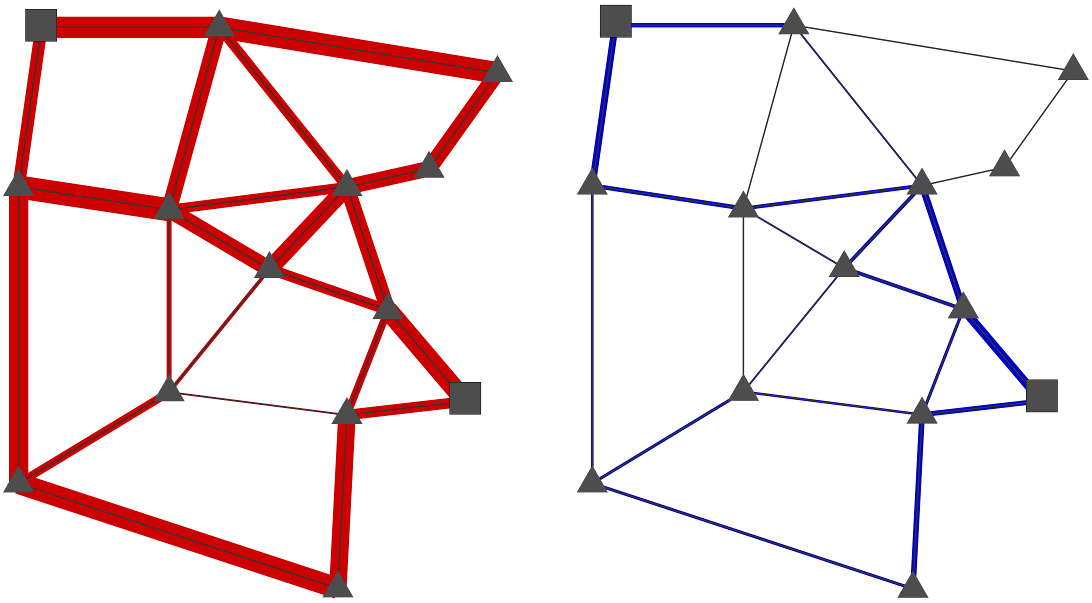


Figure 8.3: Travel time consumed by MV and AV moves in total and in platoon.

travel time in moves to lead platoons, AVs spend 85.64% of their travel time in moves as platoon followers. Beyond the values depicted in this figure, we note that a platoon is on average composed of one MV and 1.02 AVs in the solutions to this case study.

We conclude that, in the SNDMAF solutions, the MVs and AVs can be used in a more specialized way than in the MV only solutions. The MVs, that are able to travel everywhere, are used to travel between locations and pull AVs in platoons between those locations. The AVs, that are restricted in their ability to travel between locations, are used to expand the transportation capacity of MVs in platoons on selected arcs. Also, the AVs allow to hold commodities at locations until the times they are due to arrive or depart.

To further assess the effect of the heterogeneous infrastructure, we investigate the spatial distribution of the MV and AV services within the city. To this end, we illustrate the frequency of MV and AV services on the arcs of the physical network in Figure 8.4. The width of an arc corresponds to the number of services on this arc, aggregated over both directions, the complete planning horizon, and all demand instances. We distinguish between MV services in Figure 8.4a and AV services in Figure 8.4b.



(a) MV services (b) AV services

Figure 8.4: Frequency of services in the Braunschweig network, adapted from Scherr et al. (2020).

First, these figures confirm the previous finding that, in general, MVs are used more extensively for traveling between locations than AVs. From the spatial distribution of the MV services, we also notice that MVs travel through the arterial roads that lead across the city and on the roads at the periphery instead of on the ring road. Contrarily, AVs rarely travel on peripheral roads other than on the roads near the two external zones. On these roads, AVs are pulled into and out of the city center in platoons. Once the AVs are located at a satellite on the ring road, they can reposition independently on the ring road resembling mobile warehouses that perform two tasks. First, the AVs can be pulled by different MVs in platoons to those satellites that receive deliveries. Second, the respective commodities loaded on the AVs can be transshipped to MVs, which then perform the delivery.

8.2.4 Operational solutions

After investigating the tactical plans obtained from the SNDMAF in the previous sections, we now focus on the operationalization of those tactical plans. To this end, we solve the post-processing model as formulated in Chapter 7 to obtain operational solutions based on the SNDMAF solutions provided by the 2-DDD algorithm. Since the post-processing model considers only the fraction of the time-expanded network that is necessary to depict the SNDMAF solution, Gurobi is able to solve all instances to optimality within one second of runtime. As the objective of the post-processing model is to minimize the transshipment effort, we particularly investigate the assignment of commodities to individual vehicles in the following.

We first investigate the number of transshipments compared to the overall transshipment opportunities. We define these transshipment opportunities as the number of intermediary satellites a commodity passes through on its path from origin to destination according to the SNDMAF solution. In Figure 8.5, we depict the average number of transshipments per commodity as a black bar for each demand instance. Also, we depict the number of satellites at which the commodity is not transshipped as a gray bar. In this way, the overall size of each bar depicts the transshipment opportunities per commodity in a demand instance.

Observing the overall size of the bars, we first note that a commodity has 4.71 transshipment opportunities on average. This value ranges from 3.89 to 5.44 transshipment opportunities among the ten demand instances. On average over all demand instances, a commodity is only transshipped 0.50 times. Consequently, transshipments are conducted at 10.88% of the potential transshipment opportunities at intermediary satellites. However, the transshipment effort also varies between the demand instances. While the commodities in Instance 8 are transshipped 1.15 times on average, Instance 10 only requires 0.22 average transshipments per commodity.

To investigate the implementation of the commodity paths in the operational solution, we further analyze the transshipments of individual commodities. For this reason, we

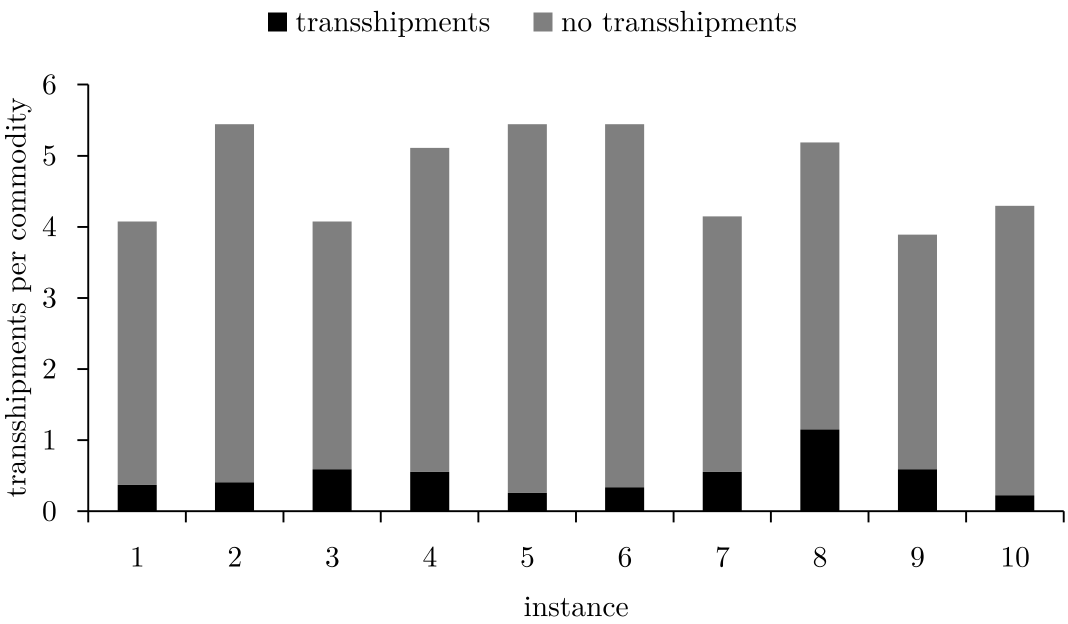


Figure 8.5: Transshipments per commodity in every demand instance.

illustrate with the histogram in Figure 8.6 how many times the commodities are transshipped on their path in all demand instances. The bars depict the share of commodities that require a respective number of transshipments. We observe that 58.89% of the commodities can be shipped directly without any transshipment between vehicles. Thus, the majority of commodities is assigned to only one vehicle on their respective path defined in the tactical plan. Around one third of the commodities require one transshipment between two vehicles. Based on the depicted values, 7.41% of the commodities require more than one transshipment to arrive at their destination.

Accomplishing the consolidations prescribed in the SNDMAF solution with minimal transshipment effort requires an appropriate selection of the individual vehicle routes. Therefore, we briefly investigate the individual MV and AV routes determined in the solution to the post-processing model. According to the results reported previously in Figure 8.3, the majority of AV services are performed in platoons. In the operational solutions, in which individual AVs are assigned to specific platoons, we further observe

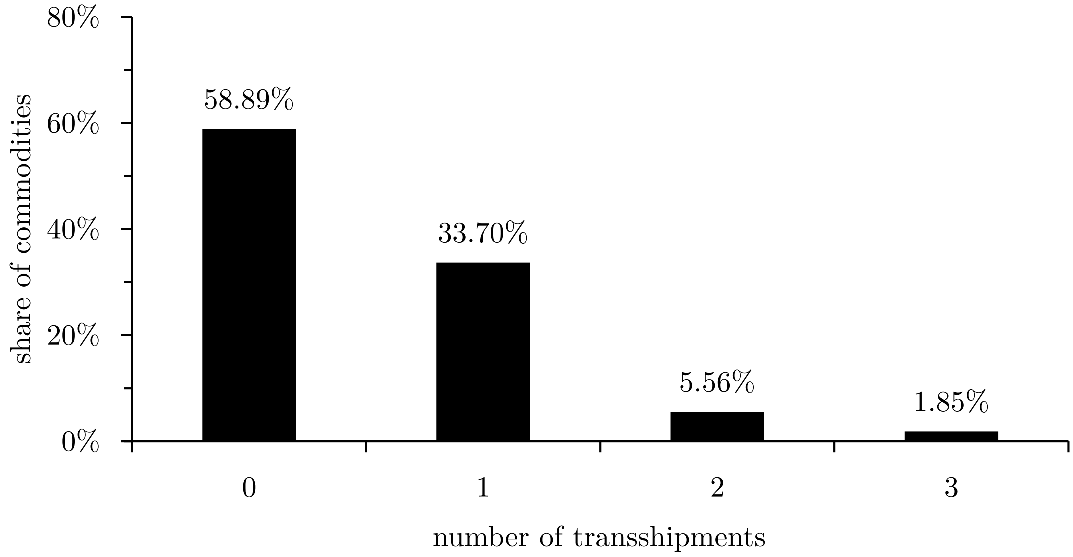


Figure 8.6: Share of commodities with different number of transshipments.

that AVs switch between different platoons over the course of the planning horizon. We note that each AV follows 1.67 different MVs on average, which represents 79.37% of the MVs that are assigned to the fleet in total. From those results, we conclude that the flexibility of AVs to switch between platoons contributes to achieving a high level of consolidation while reducing the transshipment of commodities.

8.3 Implications

We draw the following conclusions from the findings in this case study. First, we note that the proposed DDD-SNDMAF algorithms are able to solve the real-world sized instances based on the road network of Braunschweig. With the two-phase DDD-SNDMAF as an exact algorithm, the runtime can be reduced and the solution quality can be improved compared to Gurobi. Applying the heuristic search space restriction to the two-phase DDD-SNDMAF can further reduce the runtimes. For the instances in the case study, the heuristic algorithm even achieves better solutions than the exact algorithms within the runtime limit.

From the tactical plans determined for this case study, we observe an impact of using a mixed autonomous fleet in a real-world setting on the assigned vehicles to the fleet and the potential cost savings from deploying AVs. We further note that the heterogeneous infrastructure, in which the ring road enables automated driving, leads the LSP to use MVs and AVs for different tasks in the delivery process. We also evaluated the previously introduced methodology for operationalizing tactical plans. The solutions to the post-processing model show that the consolidations prescribed in the SNDMAF solutions can be implemented with reasonable transshipment effort at an operational level.

From the case study in this chapter and the experiments presented in Chapter 5, we derive the following practical implications regarding the use of mixed autonomous fleets and platooning in a city logistics context. First, we note that different types of heterogeneous infrastructure imply different fleet mixes and different strategies to utilize the vehicles of the fleet. From this, we conclude that the SNDMAF is an instrument that supports LSPs in evaluating the effectiveness of deploying a mixed autonomous on the underlying infrastructure. On the other hand, urban planning and traffic management can derive insights from the tactical plans of fleet operators, such as LSPs, as well. For example, the parts of the network may be identified that are frequented the most by AVs for contributing to cost-efficient tactical plans. Those insights may also support infrastructure decisions such as permitting or preparing roads for automated driving.

We further note that the use of AVs does not lead to a considerable increase in the overall travel time consumed by the LSP's fleet in the city. Thus, we cannot observe a negative impact of mixed autonomous fleets on traffic in the city. Platooning, which enables vehicles to follow each other with less headway, may additionally reduce the use of road space compared to individual vehicles. We further observe from the results that platooning is only used in some parts of the network and the platoons mostly consist of not more than two vehicles. This shows that potential restrictions on the length of platoons, which traffic management may impose to reduce the traffic complexity, would not necessarily compromise the LSPs' efficiency in providing their services.

Chapter 9

Conclusions

The objective of the research presented in this thesis was to analyze the opportunities and limitations from considering mixed autonomous fleets and platooning in city logistics. To this end, we proposed an optimization problem for tactical planning of an LSP, namely the service network design problem with mixed autonomous fleets (SNDMAF). We formulated a mathematical model for this tactical problem as an integer program. To solve the SNDMAF efficiently, we proposed exact and heuristic solution approaches based on the dynamic discretization discovery scheme. Further, we provided a post-processing model for transferring a tactical plan to an operational solution. The conducted experiments showed the value of considering a mixed autonomous fleet and platooning. In Section 9.1, we summarize the contributions presented in this thesis. Finally, we discuss those contributions and derive perspectives for future research in Section 9.2.

9.1 Contributions

In this thesis, we provided the following scientific contributions. In Chapter 3, we introduced the SNDMAF as a novel optimization problem for the tactical planning in city logistics. In this problem, we distinguish between two types of vehicles that compose the mixed autonomous fleet of an LSP, MVs and AVs. We further consider a heterogeneous infrastructure, in which AVs are only able to travel independently on a subset of the arcs in the network, named AV arcs. Outside of those AV arcs, they are required to follow MVs in platoons. The SNDMAF seeks to minimize the total costs of the LSP. The resulting tactical plan determines the fleet size and the mix between MVs and AVs, the scheduling of transportation services including the coordination of platoons, and the routing of commodities through the service network.

We presented a model for this problem as an integer program in Chapter 4. Additionally, we provided modifications to this model to account for different operational policies an LSP may impose. In Chapter 5, we studied different dimensions of the problem by solving instances of the model with a commercial solver. Specifically, we investigated the impact of the type of heterogeneous infrastructure, the demand pattern of commodities, and the fixed cost ratio between MVs and AVs on the solutions to the SNDMAF. These problem characteristics showed an effect on the number of MVs that can be substituted with AVs, which consequently determines the potential cost savings for the LSP. We also found that the type of heterogeneous infrastructure implies different strategies for the use of AVs and platooning.

In Chapter 6, we proposed solution approaches to efficiently solve larger instances of the SNDMAF with fine time discretization. To this end, we developed an exact algorithm based on the dynamic discretization discovery scheme that accounts for resource management considerations and the requirements of the mixed fleet autonomous fleet. Our algorithm considers two additional properties of the partially time-expanded networks, valid inequalities to strengthen the lower bound, and a new type of sparse time-expanded network for obtaining feasible solutions. We proposed two enhancements, i.e., a two-phase and a partially-relaxed variant, that use relaxations to determine optimal solutions quicker. In our computational experiments, all variants outperformed the commercial solver, with the two-phase variant providing the best performance overall. We further proposed a heuristic search space restriction that contributes to finding heuristic solutions of high quality in even shorter runtime.

In Chapter 7, we considered the operationalization of a tactical plan. Therefore, we presented the respective operational problem setting and proposed a post-processing model to transfer an SNDMAF solution to an operational solution. The model determines feasible routes for every individual MV and AV including their potential assignment to platoons. The commodities are assigned to the individual vehicles to minimize the transshipment effort.

Finally, we presented a case study in Chapter 8 to assess the deployment of a mixed autonomous fleet in a real-world environment. Thus, we considered the road network and heterogeneous infrastructure of Braunschweig. The computational results showed the ability of our solution approach to solve real-world sized instances. Overall, we assessed that a certain number of the MVs, that would be required in a conventional setting using a homogeneous fleet, can be substituted with the same number of AVs when using a mixed autonomous fleet. We also observed that the MVs and AVs are used for different tasks in the given heterogeneous infrastructure. From the solutions to the post-processing model, we concluded that the tactical plans can be implemented at the operational level with a reasonable number of transshipments.

9.2 Perspectives

From the contributions and findings presented in this thesis, we derive the following perspectives for future research and practical applications. In the problem considered in this thesis, we assume a deterministic setting in which, e.g., the commodity demand and the travel times are known. This motivates further studies that assess the effect of uncertainty, which typically occurs in the real world, on the tactical plans. Also, the SNDMAF model may be extended to recognize uncertainty for designing robust tactical plans. A two-stage formulation may consider stochastic commodity volumes or travel times.

The specific tactical planning problem of an LSP, on which we focus in this thesis, also provides perspectives for extended or adjacent problem settings. Specifically, the design of the heterogeneous infrastructure at the strategic level of planning provides an interesting area for future research. To determine a desirable heterogeneous infrastructure for the LSP, the selection of AV arcs may be included in the model. In this way, the road sequences that should be prepared for automated driving can be determined. Although one LSP may not be able to directly decide on modifications of public infrastructure, the insights gained from one or several LSPs may influence decisions of the municipality.

Another aspect for future research is the cooperation among multiple service providers to form shared platoons. The results of our studies show that LSPs are not expected to exhaust the potential maximum length of their platoons often. The excess platoon capacity may be offered to other logistics or mobility service providers in exchange for a compensation fee. Conversely, an LSP that does not have the necessary number of MVs available to platoon AVs may request additional platoon capacities from other service providers. The decisions about offering or requesting platoon capacities to form shared platoons can be integrated in an extended version of the SNDMAF model. However, cooperation requires dependability among the participating service providers to comply with the tactical plans agreed upon. Thus, game-theoretic studies on the potential incentives of individual service providers to break a cooperation should be conducted. Also, new business models may emerge from offering platoon capacity as a service to fleet operators that focus on AVs only.

In this thesis, we chose a service network design approach to model the tactical planning problem, which only considers a selection of operational considerations. Other modeling approaches such as vehicle routing problem formulations may be studied for related problem settings in future research, in particular if more operational details need to be considered. For transforming a tactical plan to an operational plan, more advanced post-processing techniques may be developed that recognize additional features such as the time needed to transfer AVs between platoons or to transship commodities.

The solution approaches for the SNDMAF that we presented in this thesis may be further improved. Therefore, more substantial modifications of the DDD scheme are required. One direction for future research is the removal of nodes and arcs from partially time-expanded networks if they are no longer required for obtaining high-quality solutions. The basis for our solution approaches, the dynamic discretization discovery scheme, relies on a variable discretization of time. This concept may be transferred to the spatial dimension of the time-expanded networks as well, in a way that the spatial resolution is variable. Applied to the SNDMAF, this would, for example, enable a more precise depiction of potential locations that allow for conducting platoon operations. Since our proposed solution approaches are relatively general, their adaptation to solve related service network design problems with resource management considerations may also be studied.

The work presented in this thesis is part of the DFG research training group SocialCars that focuses on cooperative (de-)centralized traffic management (SocialCars, 2020). In this regard, we consider a centralized approach from the perspective of one fleet operator, more precisely a city logistics service provider. Decentralized approaches for similar problem settings should also be considered and may be particularly suitable for related problem settings with multiple LSPs that are not coordinated in a centralized way. The overall objective of the project is to nudge individual traffic participants, regardless of whether they are coordinated in a centralized or decentralized way, towards improving the city's traffic efficiency and the residents' quality of life. To this end, the tactical planning approach considered in this thesis may be integrated with the corresponding planning level of traffic management in future research.

Eventually, there are preconditions that still need to be satisfied before mixed autonomous fleets can be successfully deployed in real-world city logistics applications. The acceptance of automated vehicles among the public is critical to overcome regulatory burdens that exist for now. While improving the safety of automated vehicles is important, the economic effects of automation and their perception also pose challenges for the sustainable deployment of automated driving. This motivates further research projects focusing on future traffic concepts and their application.

Bibliography

Agatz, N., Bazzan, A. L. C., Kutadinata, R., Mattfeld, D. C., Sester, M., Winter, S., and Wolfson, O. (2016). Autonomous car and ride sharing: flexible road trains. *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '16*, pages 1–4.

Agatz, N., Bouman, P., and Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, 52(4):965–981.

Andersen, J., Christiansen, M., Crainic, T. G., and Grønhaug, R. (2011). Branch and price for service network design with asset management constraints. *Transportation Science*, 45(1):33–49.

Andersen, J., Crainic, T. G., and Christiansen, M. (2009). Service network design with asset management: Formulations and comparative analyses. *Transportation Research Part C: Emerging Technologies*, 17(2):197–207.

Axelsson, J. (2017). Safety in vehicle platooning: A systematic literature review. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1033–1045.

Barnhart, C., Krishnan, N., Kim, D., and Ware, K. (2002). Network design for express shipment delivery. *Computational Optimization and Applications*, 21(3):239–262.

Barrachina, J., Garrido, P., Fogue, M., Martinez, F. J., Cano, J. C., Calafate, C. T., and Manzoni, P. (2013). Road side unit deployment: A density-based approach. *IEEE Intelligent Transportation Systems Magazine*, 5(3):30–39.

Belieres, S. (2019). *Mathematical programming for tactical transportation planning in a multi-product supply chain*. PhD thesis, INSA de Toulouse.

Benjelloun, A., Crainic, T. G., and Bigras, Y. (2010). Towards a taxonomy of city logistics projects. *Procedia - Social and Behavioral Sciences*, 2(3):6217–6228.

Bergenhem, C., Shladover, S., Coelingh, E., Englund, C., and Tsugawa, S. (2012). Overview of platooning systems. In *Proceedings of the 19th ITS World Congress, Oct 22-26, Vienna, Austria (2012)*.

Bhoopalam, A. K., Agatz, N., and Zuidwijk, R. (2018). Planning of truck platoons: A literature review and directions for future research. *Transportation Research Part B: Methodological*, 107:212–228.

Boland, N., Hewitt, M., Marshall, L., and Savelsbergh, M. (2017). The continuous-time service network design problem. *Operations Research*, 65(5):1303–1321.

Boland, N., Hewitt, M., Marshall, L., and Savelsbergh, M. (2019). The price of discretizing time: a study in service network design. *EURO Journal on Transportation and Logistics*, 8(2):195–216.

Boysen, N., Briskorn, D., and Schwerdfeger, S. (2018a). The identical-path truck platooning problem. *Transportation Research Part B: Methodological*, 109:26–39.

Boysen, N., Schwerdfeger, S., and Weidinger, F. (2018b). Scheduling last-mile deliveries with truck-based autonomous robots. *European Journal of Operational Research*, 271(3):1085–1099.

Bukszpan, D. (2019). Here's how Amazon robots could make the deliveryman extinct – CNBC. `https://www.cnbc.com/2019/04/26/heres-how-amazon-robots-could-make-the-deliveryman-extinct.html`. Last accessed on September 23, 2020.

CARMERA (2018). Against The Odds: Why The AV Industry Should Probably Focus On Just The Even-Numbered SAE Autonomy Levels. `https:`

//medium.com/field-of-view/against-the-odds-why-the-av-industry-should-probably-focus-on-just-the-even-numbered-sae-autonomy-ed4510633dcf. Last accessed on September 23, 2020.

Cattaruzza, D., Absi, N., Feillet, D., and González-Feliu, J. (2017). Vehicle routing problems for city logistics. *EURO Journal on Transportation and Logistics*, 6(1):51–79.

Chan, E. (2016). Sartre automated platooning vehicles. In *Towards Innovative Freight and Logistics*, volume 2, chapter 10, pages 137–150. John Wiley & Sons, Ltd.

Chen, Z., He, F., Yin, Y., and Du, Y. (2017). Optimal design of autonomous vehicle zones in transportation networks. *Transportation Research Part B: Methodological*, 99:44–61.

Chottani, A., Hastings, G., Murnane, J., and Neuhaus, F. (2018). Distraction or disruption? Autonomous trucks gain ground in US logistics – McKinsey. https://www.mckinsey.com/industries/travel-logistics-and-transport-infrastructure/our-insights/distraction-or-disruption-autonomous-trucks-gain-ground-in-us-logistics. Last accessed on September 23, 2020.

Chouman, M. and Crainic, T. G. (2015). Cutting-plane matheuristic for service network design with design-balanced requirements. *Transportation Science*, 49(1):99–113.

Coker, A. (2018). Lyft looks to create human, autonomous hybrid network in ride-sharing space – FreightWaves. https://www.freightwaves.com/news/lyft-looks-to-create-human-autonomous-hybrid-network-in-ride-sharing-space. Last accessed on September 23, 2020.

Coren, M. J. (2017). The world's first self-driving "rail bus" without tracks is taking its inaugural journey in China – Quartz. https://qz.com/1125244/the-worlds-first-self-driving-rail-bus-without-tracks-is-taking-its-inaugural-journey-in-china/. Last accessed on September 23, 2020.

Crainic, T. G. (2000). Service network design in freight transportation. *European Journal of Operational Research*, 122(2):272–288.

Crainic, T. G. (2008). City logistics. In *State-of-the-Art Decision-Making Tools in the Information-Intensive Age*, pages 181–212. INFORMS.

Crainic, T. G., Errico, F., Rei, W., and Ricciardi, N. (2012). Integrating c2e and c2c traffic into city logistics planning. *Procedia - Social and Behavioral Sciences*, 39:47–60.

Crainic, T. G., Frangioni, A., and Gendron, B. (2001). Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics*, 112(1-3):73–99.

Crainic, T. G., Gendreau, M., and Jemai, L. (2020). Planning hyperconnected, urban logistics systems. *Transportation Research Procedia*, 47:35–42.

Crainic, T. G., Hewitt, M., Toulouse, M., and Vu, D. M. (2016). Service network design with resource constraints. *Transportation Science*, 50(4):1380–1393.

Crainic, T. G., Hewitt, M., Toulouse, M., and Vu, D. M. (2017). Scheduled service network design with resource acquisition and management. *EURO Journal on Transportation and Logistics*, 7(3):277–309.

Crainic, T. G. and Laporte, G. (1997). Planning models for freight transportation. *European Journal of Operational Research*, 97(3):409–438.

Crainic, T. G. and Montreuil, B. (2016). Physical internet enabled hyperconnected city logistics. *Transportation Research Procedia*, 12:383–398.

Crainic, T. G., Perboli, G., Mancini, S., and Tadei, R. (2010). Two-echelon vehicle routing problem: A satellite location analysis. *Procedia - Social and Behavioral Sciences*, 2(3):5944–5955.

Crainic, T. G., Ricciardi, N., and Storchi, G. (2009). Models for evaluating and planning city logistics systems. *Transportation Science*, 43(4):432–454.

Crainic, T. G. and Rousseau, J.-M. (1986). Multicommodity, multimode freight transportation: A general modeling and algorithmic framework for the service network design problem. *Transportation Research Part B: Methodological*, 20(3):225 – 242.

Crainic, T. G. and Sgalambro, A. (2014). Service network design models for two-tier city logistics. *Optimization Letters*, 8(4):1375–1387.

Dash, S., Günlük, O., Lodi, A., and Tramontani, A. (2012). A time bucket formulation for the traveling salesman problem with time windows. *INFORMS Journal on Computing*, 24(1):132–147.

Davies, A. (2019). This Robo-Van Startup Will Handle Walmart's 'Middle Mile' – WIRED. `https://www.wired.com/story/robo-van-startup-handle-walmarts-middle-mile/`. Last accessed on September 23, 2020.

Derigs, U., Pullmann, M., and Vogel, U. (2013). Truck and trailer routing – problems, heuristics and computational experience. *Computers and Operations Research*, 40(2):536–546.

Duncan, I. (2020). Self-driving car company Nuro wins federal safety approval for grocery delivery robot – The Washington Post. `https://www.washingtonpost.com/local/trafficandcommuting/grocery-delivery-robot-wins-first-federal-safety-approval-for-a-self-driving-vehicle/2020/02/06/90e97658-48f5-11ea-9164-d3154ad8a5cd_story.html`. Last accessed on September 23, 2020.

Ehmke, J. F., Steinert, A., and Mattfeld, D. C. (2012). Advanced routing for city logistics service providers based on time-dependent travel times. *Journal of computational science*, 3(4):193–205.

Englund, C., Chen, L., Ploeg, J., Semsar-Kazerooni, E., Voronov, A., Bengtsson, H. H., and Didoff, J. (2016). The grand cooperative driving challenge 2016: boosting the introduction of cooperative automated vehicles. *IEEE Wireless Communications*, 23(4):146–152.

European Automobile Manufacturers Association (2017). EU Roadmap for Truck Platooning. `https://www.acea.be/uploads/publications/Platooning_roadmap.pdf`. Last accessed on September 23, 2020.

Fleischer, L. and Skutella, M. (2007). Quickest flows over time. *SIAM Journal on Computing*, 36(6):1600–1630.

Fontaine, P., Crainic, T. G., Fontaine, P., Crainic, T. G., Jabali, O., and Rei, W. (2017). Multi-modal scheduled service network design with resource management for two-tier city logistics. *CIRRELT Working Paper*, 2017(27).

Ford, L. R. and Fulkerson, D. R. (1958). Constructing maximal dynamic flows from static flows. *Operations Research*, 6(3):419–433.

Forschungsgesellschaft für Straßen- und Verkehrswesen (2010). Richtlinien für Lichtsignalanlagen (RiLSA).

Franceschetti, A., Honhon, D., Laporte, G., Woensel, T. V., and Fransoo, J. C. (2017). Strategic fleet planning for city logistics. *Transportation Research Part B: Methodological*, 95:19–40.

Gianessi, P., Alfandari, L., Létocart, L., and Calvo, R. W. (2016). The multicommodity-ring location routing problem. *Transportation Science*, 50(2):541–558.

Gnegel, F. and Fügenschuh, A. (2020). An iterative graph expansion approach for the scheduling and routing of airplanes. *Computers and Operations Research*, 114:104832.

Gurobi Optimization (2020). Gurobi Optimizer. `https://www.gurobi.com/products/gurobi-optimizer/`. Last accessed on September 23, 2020.

He, E., Boland, N., Nemhauser, G., and Savelsbergh, M. (2018). A dynamic discretization discovery algorithm for the minimum duration time-dependent shortest path problem. In van Hoeve, W.-J., editor, *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 289–297. Springer International Publishing.

Hemmelmayr, V. C., Cordeau, J. F., and Crainic, T. G. (2012). An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers and Operations Research*, 39(12):3215–3228.

Heutger, M. and Kückelhaus, M. (2014). Self-Driving Vehicles in Logistics – DHL Trend Research. `http://www.dhl.com/content/dam/downloads/g0/about_us/logistics_insights/dhl_self_driving_vehicles.pdf`. Last accessed on September 23, 2020.

Hewitt, M. (2019). Enhanced Dynamic Discretization Discovery for the Continuous Time Load Plan Design Problem. *Transportation Science*, 53(6):1731–1750.

Hewitt, M., Crainic, T. G., Nowak, M., and Rei, W. (2019). Scheduled service network design with resource acquisition and management under uncertainty. *Transportation Research Part B: Methodological*, 128:324–343.

Hook, L. (2018). Driverless cars: mapping the trouble ahead – Financial Times. `https://www.ft.com/content/2a8941a4-1625-11e8-9e9c-25c814761640`. Last accessed on September 23, 2020.

IBM (2020). IBM ILOG CPLEX Optimization Studio – IBM. `https://www.ibm.com/products/ilog-cplex-optimization-studio`. Last accessed on September 23, 2020.

Johansson, A. and Martensson, J. (2019). Game theoretic models for profit-sharing in multi-fleet platoons. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3019–3024. IEEE.

Lagos, F., Boland, N., and Savelsbergh, M. (2020). The continuous-time inventory-routing problem. *Transportation Science*, 54(2):375–399.

Lammert, M. P., Duran, A., Diez, J., Burton, K., and Nicholson, A. (2014). Effect of platooning on fuel consumption of class 8 vehicles over a range of speeds, following distances, and mass. *SAE International Journal of Commercial Vehicles*, 7(2):626–639.

Larsen, R., Rich, J., and Rasmussen, T. K. (2019). Hub-based truck platooning: Potentials and profitability. *Transportation Research Part E: Logistics and Transportation Review*, 127:249–264.

Larson, J., Munson, T., and Sokolov, V. (2016). Coordinated platoon routing in a metropolitan network. In *2016 Proceedings of the Seventh SIAM Workshop on Combinatorial Scientific Computing*, pages 73–82. SIAM.

Larsson, E., Sennton, G., and Larson, J. (2015). The vehicle platooning problem: Computational complexity and heuristics. *Transportation Research Part C: Emerging Technologies*, 60:258–277.

Li, Y., Chen, Z., Yin, Y., and Peeta, S. (2020). Deployment of roadside units to overcome connectivity gap in transportation networks with mixed traffic. *Transportation Research Part C: Emerging Technologies*, 111:496–512.

Lioris, J., Pedarsani, R., Tascikaraoglu, F. Y., and Varaiya, P. (2016). Doubling throughput in urban roads by platooning. *IFAC-PapersOnLine*, 49(3):49–54.

Lo, H. K., An, K., and hua Lin, W. (2013). Ferry service network design under demand uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 59:48–70.

Magnanti, T. L. and Wong, R. T. (1984). Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55.

Mahmassani, H. S. (2016). 50th anniversary invited article – Autonomous vehicles and connected vehicle systems: Flow and operations considerations. *Transportation Science*, 50(4):1140–1162.

Maiti, S., Winter, S., and Kulik, L. (2017). A conceptualization of vehicle platoons and platoon operations. *Transportation Research Part C: Emerging Technologies*, 80:1–19.

Masoud, N. and Jayakrishnan, R. (2017). Autonomous or driver-less vehicles: Implementation strategies and operational concerns. *Transportation Research Part E: Logistics and Transportation Review*, 108:179–194.

Masson, R., Trentini, A., Lehuédé, F., Malhéné, N., Péton, O., and Tlahig, H. (2017). Optimization of a city logistics transportation system with mixed passengers and goods. *EURO Journal on Transportation and Logistics*, 6(1):81–109.

McLane, R. (2019). PlatoonPro Truck Platooning – Peloton Technology. `https://peloton-tech.com/wp-content/uploads/2019/03/safety_report_updated.pdf`. Last accessed on September 23, 2020.

Medina, J., Hewitt, M., Lehuédé, F., and Péton, O. (2019). Integrating long-haul and local transportation planning: the service network design and routing problem. *EURO Journal on Transportation and Logistics*, 8(2):119–145.

Meisel, F. and Kopfer, H. (2014). Synchronized routing of active and passive means of transport. *OR Spectrum*, 36(2):297–322.

Mosquet, X., Dauner, T., Lang, N., Rübmann, M., Mei-Pochtler, A., Agrawal, R., and Schmieg, F. (2015). Revolution in the Driver's Seat – Boston Consulting Group. `https://www.bcgperspectives.com/content/articles/automotive-consumer-insight-revolution-drivers-seat-road-autonomous-vehicles/`. Last accessed on September 23, 2020.

Murray, C. C. and Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54:86–109.

Neumann-Saavedra, B. A., Crainic, T. G., Gendron, B., Mattfeld, D. C., and Römer, M. (2020). Integrating resource management in service network design for bike-sharing systems. *Transportation Science*, 54(5):1251–1271.

Nothdurft, T., Hecker, P., Ohl, S., Saust, F., Maurer, M., Reschka, A., and Böhmer, J. R. (2011). Stadtpilot: First fully autonomous test drives in urban traffic. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 919–924. IEEE.

Parker, N., Shandro, A., and Cullen, E. (2017). Autonomous and connected vehicles: navigating the legal issues. Technical report, Allen & Overy.

Pedersen, M. B., Crainic, T. G., and Madsen, O. B. G. (2009). Models and tabu search metaheuristics for service network design with asset-balance requirements. *Transportation Science*, 43(2):158–177.

Persiel, S. (2017). DHL und Deutsche Post in Braunschweig. `https://www.paketda.de/dhl-deutsche-post-braunschweig.html`. Last accessed on September 23, 2020.

Poikonen, S. and Golden, B. (2019). Multi-visit drone routing problem. *Computers and Operations Research*, 113:104802.

Poikonen, S., Wang, X., and Golden, B. (2017). The vehicle routing problem with drones: Extended models and connections. *Networks*, 70(1):34–43.

Renault (2018). EZ-Pro Concept. `http://fr.zone-secure.net/-/Groupe_Renault/Dossier_Presse_EZ_PRO_EN/-/`. Last accessed on September 23, 2020.

SAE International (2018). *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles.*

Savelsbergh, M. and Van Woensel, T. (2016). 50th anniversary invited article – City logistics: Challenges and opportunities. *Transportation Science*, 50(2):579–590.

Scania Group (2018). Automated platooning – step by step. `https://www.scania.com/group/en/home/newsroom/news/2018/automated-platooning-step-by-step.html`. Last accessed on September 23, 2020.

Scherr, Y. O., Hewitt, M., Neumann Saavedra, B. A., and Mattfeld, D. C. (2020). Dynamic discretization discovery for the service network design problem with mixed autonomous fleets. *Transportation Research Part B: Methodological*, 141:164–195.

Scherr, Y. O., Neumann-Saavedra, B. A., Hewitt, M., and Mattfeld, D. C. (2018). Service network design for same day delivery with mixed autonomous fleets. *Transportation Research Procedia*, 30:23–32.

Scherr, Y. O., Neumann Saavedra, B. A., Hewitt, M., and Mattfeld, D. C. (2019). Service network design with mixed autonomous fleets. *Transportation Research Part E: Logistics and Transportation Review*, 124:40–55.

Schmeitz, A., Schwartz, R. S., Ravesteijn, D., Verhaeg, G., Altgassen, D., and Wedemeijer, H. (2019). EU AUTOPILOT project: Platooning use case in Brainport. In *13th ITS European Congress, Brainport, the Netherlands, 3-6 June 2019*.

Sebe, S. M., Kraus, P., Müller, J. P., and Westphal, S. (2019). Cross-provider platoons for same-day delivery. In *VEHITS 2019 - Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems*, pages 106–116.

Shladover, S. E. (2007). PATH at 20 – History and major milestones. *IEEE Transactions on intelligent transportation systems*, 8(4):584–592.

Shladover, S. E., Nowakowski, C., Lu, X.-Y., and Ferlis, R. (2015). Cooperative adaptive cruise control: Definitions and operating concepts. *Transportation Research Record: Journal of the Transportation Research Board*, 2489(1):145–152.

Skutella, M. (2009). An introduction to network flows over time. In Cook, W., Lovász, L., and Vygen, J., editors, *Research Trends in Combinatorial Optimization*, pages 451–482. Springer Berlin Heidelberg.

SocialCars (2020). Cooperative (de-)centralized traffic management – DFG Research Training Group 1931. `https://www.socialcars.org/home.html`. Last accessed on September 23, 2020.

Starship Technologies (2020). Starship. https://www.starship.xyz/. Last accessed on September 23, 2020.

Stopard, R., Plant, A., and Miles, J. (2015). Affordable Mass Transit for Cambridge and the Wider Region. https://www.connectingcambridgeshire.co.uk/wp-content/uploads/2018/07/Affordable-Very-Rapid-Transit-AVRT-Report.-Vol-1..pdf. Last accessed on September 23, 2020.

Talebpour, A. and Mahmassani, H. S. (2016). Influence of connected and autonomous vehicles on traffic flow stability and throughput. *Transportation Research Part C: Emerging Technologies*, 71:143–163.

Taniguchi, E. and Thompson, R. (2001). *City Logistics: Network Modelling and Intelligent Transport Systems*. Emerald Group Publishing Limited.

Taniguchi, E. and Thompson, R. G. (2002). Modeling city logistics. *Transportation research record*, 1790(1):45–51.

Taniguchi, E. and Thompson, R. G. (2014). *City logistics: Mapping the future*. CRC Press.

Tsugawa, S., Kato, S., and Aoki, K. (2011). An automated truck platoon for energy saving. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4109–4114. IEEE.

Ulmer, M. W. and Streng, S. (2019). Same-day delivery with pickup stations and autonomous vehicles. *Computers and Operations Research*, 108:1–19.

VDA (2020). Platooning – Innovation im Güterverkehr - VDA. https://www.vda.de/de/themen/innovation-und-technik/automatisiertes-fahren/platooning.html. Last accessed on September 23, 2020.

Vigo, D. and Toth, P. (2014). *Vehicle Routing: Problems, Methods, and Applications, Second Edition*. SIAM.

Volkswagen AG (2018). Platooning in the logistics industry: world's first practical use of networked truck convoys in Germany. `https://www.volkswagenag.com/en/news/2018/06/man_start_platooning.html`. Last accessed on September 23, 2020.

Volvo Trucks USA (2018). Volvo Trucks and FedEx Successfully Demonstrate Truck Platooning on NC 540. `https://www.volvotrucks.us/news-and-stories/press-releases/2018/june/volvo-trucks-and-fedex-successfully-demonstrate-truck-platooning-on-nc-540/`. Last accessed on September 23, 2020.

Vu, D. M., Hewitt, M., Boland, N., and Savelsbergh, M. (2020). Dynamic discretization discovery for solving the time dependent traveling salesman problem with time windows. *Transportation Science*, 54(3):703–720.

Wang, M., van Maarseveen, S., Happee, R., Tool, O., and van Arem, B. (2019). Benefits and risks of truck platooning on freeway operations near entrance ramp. *Transportation Research Record: Journal of the Transportation Research Board*, 2673(8):588–602.

Wang, X. and Regan, A. C. (2002). Local truckload pickup and delivery with hard time window constraints. *Transportation Research Part B: Methodological*, 36(2):97–112.

Wang, X. and Regan, A. C. (2009). On the convergence of a new time window discretization method for the traveling salesman problem with time window constraints. *Computers and Industrial Engineering*, 56(1):161–164.

Wang, Z. and Qi, M. (2019). Service network design considering multiple types of services. *Transportation Research Part E: Logistics and Transportation Review*, 126:1–14.

Wang, Z. and Sheu, J. B. (2019). Vehicle routing problem with drones. *Transportation Research Part B: Methodological*, 122:350–364.

Wieberneit, N. (2008). Service network design for freight transportation: a review. *OR Spectrum*, 30(1):77–112.